THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *03/10/2019* **par :**

Léo Grange

Datacenter Management for on-site Intermittent and Uncertain Renewable Energy Sources

---

JURY

Laetitia Jourdan	Professeur	Rapporteur
Gaël Thomas	Professeur	Rapporteur
Jean-Marc Menaud	Professeur	Examinateur
Laurent Lefevre	Chargé de Recherche	Examinateur
Jean-Marc Pierson	Professeur	Examinateur
Georges Da Costa	Maître de Conférences	Co-Directeur de thèse
Patricia Stolf	Maître de Conférences	Co-Directrice de thèse
Paul Renaud-Goud	Maître de Conférences	Co-Encadrant

École doctorale et spécialité :
MITT : Domaine STIC : Sureté de logiciel et calcul de haute performance
Unité de Recherche :
IRIT (UMR 5505)
Directeur(s) de Thèse :
Patricia Stolf et *Georges Da Costa*
Rapporteurs :
Laetitia Jourdan et *Gaël Thomas*

Acknowledgments

First, I want to thank my supervisors, Georges Da Costa and Patricia Stolf for their support along these three years and for enduring me all that time, especially during the period I was writing this manuscript. Their scientific guidance and moral support were of great help. I would like to thank Paul Renaud-Goud as well, both for the various discussions on the terrace and for the great help on scientific aspects.

I would also like to thank the other members of the jury, for their reviews and comments: Laetitia Jourdan, Gaël Thomas, Jean-Marc Menaud, Laurent Lefevre and Jean-Marc Pierson.

I would like to thank Jean-Marc Pierson a second time for welcoming me in the SEPIA team and in the Datazero project. While my work diverged progressively from the view of the project, I appreciated the discussions and encouragements from the Datazero members as well as the good time spent with the other PhD students after the meetings. My sincere gratitude goes to the rest of the SEPIA team, especially my office mates. Inès and Zongyi who played the role of the elders students in the office when I arrived; Gustavo and Chaopeng who were my companions during almost the three years and with whom we exchanged a lot during breaks, lunches and the rare barbecues I went to; Tanissia who represents the new generation at our office, hoping she will gain merited self-confidence and stop experiencing Murphy's law.

I crossed paths with several other people during this research journey. A special thank is addressed to David Sanchez, who provided valuable clues on a mathematical model and actually saved our analysis. During my short visit in Poland, I met great persons at the PSNC and hope to collaborate with them in the future. Hence I would like to say dziękuję bardzo *to Wojciech, Ariel, Mateusz and Michał.*

I could not forget to thank all my friends and roommates, who were understanding of my intermittent availability and always there when I needed to relax or to try to solve world's problems in endless but exciting discussions. It is not possible to be exhaustive, but I want to mention at least Enzo, Wladimir, Lina, Milena, Étienne, Ophélie, Malik, all the Nicolas, Yohan, Guillaume, Théo and Sarah.

I want to thank my parents, Alain and Anne-Marie, as well as my brothers and sisters Miléna, Éva, Guillaume and Simon. I am very lucky of having such an amazing family, all of them being benevolent and always doing their best to support my choices and discuss of my doubts.

A special thank to Terry Pratchett, whose I unfortunately discovered after he passed away but who provided me great readings. The final though goes to the newcomer in this world which arrived during my PhD, Myria and Johan, which make us remember that the sustainability of our world is more than an abstract topic.

Some part of me can't wait to see what life's going to come up with next! Anticipation without the usual anxiety. And underneath it all is the feeling that we both belong here, just as we are, right now.

— Alexander Shulgin

Abstract

In recent years, information and communication technologies (ICT) became a major energy consumer, with the associated harmful ecological consequences. Indeed, the emergence of Cloud computing and massive Internet companies increased the importance and number of datacenters around the world. In order to mitigate economical and ecological cost, powering datacenters with renewable energy sources (RES) began to appear as a sustainable solution.

Some of the commonly used RES, such as solar and wind energies, directly depends on weather conditions. Hence they are both intermittent and partly uncertain. Batteries or other energy storage devices (ESD) are often considered to relieve these issues, but they result in additional energy losses and are too costly to be used alone without more integration. The power consumption of a datacenter is closely tied to the computing resource usage, which in turn depends on its workload and on the algorithms that schedule it. To use RES as efficiently as possible while preserving the quality of service of a datacenter, a coordinated management of computing resources, electrical sources and storage is required.

A wide variety of datacenters exists, each with different hardware, workload and purpose. Similarly, each electrical infrastructure is modeled and managed uniquely, depending on the kind of RES used, ESD technologies and operating objectives (cost or environmental impact). Some existing works successfully address this problem by considering a specific couple of electrical and computing models. However, because of this combined diversity, the existing approaches cannot be extrapolated to other infrastructures.

This thesis explores novel ways to deal with this coordination problem. A first contribution revisits batch tasks scheduling problem by introducing an abstraction of the power sources. A scheduling algorithm is proposed, taking preferences of electrical sources into account, though designed to be independent from the type of sources and from the goal of the electrical infrastructure (cost, environmental impact, or a mix of both).

A second contribution addresses the joint power planning coordination problem in a totally infrastructure-agnostic way. The datacenter computing resources and workload management is considered as a black-box implementing a scheduling under variable power constraint algorithm. The same goes for the electrical sources and storage management system, which acts as a source commitment optimization algorithm. A cooperative multi-objective power planning optimization, based on a multi-objective evolutionary algorithm (MOEA), dialogues with the two black-boxes to find the best trade-offs between electrical and computing internal objectives.

Finally, a third contribution focuses on RES production uncertainties in a more specific infrastructure. Based on a Markov Decision Process (MDP) formulation, the structure of the underlying decision problem is studied. For several variants of the problem, tractable methods are proposed to find optimal policies or a bounded approximation.

Résumé

Les technologies de l'information et de la communication sont devenues, au cours des dernières années, un pôle majeur de consommation énergétique avec les conséquences environnementales associées. Dans le même temps, l'émergence du *Cloud computing* et des grandes plateformes en ligne a causé une augmentation en taille et en nombre des centres de données. Pour réduire leur impact écologique, alimenter ces centres avec des sources d'énergies renouvelables (EnR) apparaît comme une piste de solution.

Cependant, certaines EnR telles que les énergies solaires et éoliennes sont liées aux conditions météorologiques, et sont par conséquent intermittentes et incertaines. L'utilisation de batteries ou d'autres dispositifs de stockage est souvent envisagée pour compenser ces variabilités de production. De par leur coût important, économique comme écologique, ainsi que les pertes énergétiques engendrées, l'utilisation de ces dispositifs sans intégration supplémentaire est insuffisante. La consommation électrique d'un centre de données dépend principalement de l'utilisation des ressources de calcul et de communication, qui est déterminée par la charge de travail et les algorithmes d'ordonnancement utilisés. Pour utiliser les EnR efficacement tout en préservant la qualité de service du centre, une gestion coordonnée des ressources informatiques, des sources électriques et du stockage est nécessaire.

Il existe une grande diversité de centres de données, ayant différents types de matériel, de charge de travail et d'utilisation. De la même manière, suivant les EnR, les technologies de stockage et les objectifs en termes économiques ou environnementaux, chaque infrastructure électrique est modélisée et gérée différemment des autres. Des travaux existants proposent des méthodes de gestion d'EnR pour des couples bien spécifiques de modèles électriques et informatiques. Cependant, les multiples combinaisons de ces deux parties rendent difficile l'extrapolation de ces approches et de leurs résultats à des infrastructures différentes.

Cette thèse explore de nouvelles méthodes pour résoudre ce problème de coordination. Une première contribution reprend un problème d'ordonnancement de tâches en introduisant une abstraction des sources électriques. Un algorithme d'ordonnancement est proposé, prenant les préférences des sources en compte, tout en étant conçu pour être indépendant de leur nature et des objectifs de l'infrastructure électrique.

Une seconde contribution étudie le problème de planification de l'énergie d'une manière totalement agnostique des infrastructures considérées. Les ressources informatiques et la gestion de la charge de travail sont encapsulées dans une boîte noire implémentant un ordonnancement sous contrainte de puissance. La même chose s'applique pour le système de gestion des EnR et du stockage, qui agit comme un algorithme d'optimisation d'engagement de sources pour répondre à une demande. Une optimisation coopérative et multiobjectif, basée sur un algorithme évolutionnaire, utilise ces deux boîtes noires afin de trouver les meilleurs compromis entre les objectifs électriques et informatiques.

Enfin, une troisième contribution vise les incertitudes de production des EnR pour

une infrastructure plus spécifique. En utilisant une formulation en processus de décision markovien (MDP), la structure du problème de décision sous-jacent est étudiée. Pour plusieurs variantes du problème, des méthodes sont proposées afin de trouver les politiques optimales ou des approximations de celles-ci avec une complexité raisonnable.

Contents

Chapter 1

Introduction

If knowledge can create problems, it is not through ignorance that we can solve them.

— Isaac Asimov

1.1 Context

Today more than ever, the energy consumption generated by the growing use of information and communication technologies (ICT) is a major issue, from both economical and ecological points of view. The emergence and development of grid computing and cloud computing paradigms, during the last decade, caused an increase in the number and size of datacenters. Datacenters are becoming a significant part of the global electrical consumption. Their total consumption in 2012 was estimated to almost 270 TWh [187]. This was roughly equivalent to 1.4% of the worldwide electrical consumption, while the complete ICT sector (excluding manufacturing) accounts for 4.7% of it. According to the same study, the datacenter power needs increased annually by 5% between 2006 and 2012. The projections for the next decade suggest higher growth rate in the near future. In year 2030, the datacenters alone may use between 3% (best case) and 13% (worst scenario) of the global electricity production [8].

Because of the current energy consumption of the ICT sector and its growing requirement, the responsibility of big companies in terms of greenhouse gas emission and pollution in general is often pointed out. However, ICT firms seem more involved in reducing their ecological impact than other industry sectors. Several companies such as Intel or Adobe buy enough renewable energy to cover all their consumption in United States [141]. This is also pointed out by Greenpeace, in their Click Clean report[1] claiming that, between 2011 and 2016, 16 of the major internet companies made "a meaningful long-term commitment to be 100% renewably powered". There are many reasons for this recent change in companies behaviors. Radu [156] studied the determinants of Green ICT adoption in general, pointing out economic, ethical and regulatory-related reasons, such as long term cost reduction, pro-environment grants, organization strategy and image.

The transition toward global sustainable energy is one of the important topic of this century. Krakowski et al. [107] study several scenarios for increased penetration of renewable energies in the French electrical mix. Multiple scenarios target 100% penetration in 2050, showing feasibility, cost and limitations of each. The authors also compare their

[1] «Clicking Clean: Who is winning the race to build a green internet?», http://www.clickclean.org/downloads/ClickClean2016%20HiRes.pdf

results to a previous prospective study from the french environment and energy management agency (ADEME)[2] with similar objectives but using different methods. Several agencies and researchers published similar studies for other regions, such as Elliston et al. for Australia [62]. These studies show that renewable energy at a large scale is technically and economically realistic for the next decades. However, more work should be done in order to reach this goal as soon as possible, by studying not only how to produce this energy, but also how to build systems interacting well with such renewable sources.

There are two ways to increase the renewable energy usage of a facility such as a datacenter: *off-site* or *on-site* integration. In the off-site scheme, the datacenter relies entirely on the electrical grid, therefore consuming energy from the energy mix of the area where it is installed. However, for every kW h consumed, the same amount of renewable energy is eventually bought from renewable energy sources (RES) installed in other locations, often hundreds or thousands of kilometers away and injected in the grid at different places and times. Hence the datacenter relies indistinctly on local renewable/non-renewable energies, but the operator uses the electrical grid market to somehow compensate its consumption with renewable production.

Oppositely, the *on-site* integration consists in building RES physically tied to the electrical network of the datacenter. Several operators have already built datacenters at least partially powered with on-site renewable energies. For instance AISO.net, a small cloud service provider, uses on-site solar panels to run its datacenter. More recently, in 2018, Google announced the construction of a massive solar power plant producing 2.9 GWh annually on the Saint-Ghislain datacenter facility site, in Belgium. The integration of on-site RES, to be really effective in terms of operating ecological impact and cost, requires major changes in the design of a datacenter management.

Most of the commonly used RES, like solar and wind, depends of weather conditions. Hence they are intermittent and the available power is difficult to predict due to the uncertainties in weather forecasting. A naive approach to use on-site RES to power a datacenter connected to the electrical grid is to simply buy the grid power when renewable is lower than the consumption and resell any excess of renewable on the grid. In this case, no modifications are required in the datacenter management. Two main issues are raised however. The transmission and distribution of the sold energy induces some energy losses, which count between 8% in Europe and North America but up to 17% in Latin America [92]. It is therefore more efficient to consume the power directly instead of injecting it in the electrical grid. The second issue is more serious, as it becomes more and more important with the growth of RES power plants. In a given region, much of the solar and wind sources tends to have their production peaks and dips close in time, and with their increasing share of the total electricity generation these fluctuations are causing general over- and under-production in the whole grid. This has, in turn, a direct effect on the price of the electricity [178]. Therefore, in case of excess of renewable production, the energy will be sold cheaper to the grid, and inversely the cost of the energy required when on-site RES does not produce enough will be higher.

In order to use renewable energy efficiently, a deeper integration is required. Adding energy storage devices (ESD), such as batteries or fuel cells allows to absorb some of the production peaks to use the energy latter, effectively smoothing the renewable power. But all ESD have a finite capacity, an imperfect efficiency, additional cost and are often subject to aging depending on their usage. They additionally require a more advanced management system to take care of their charge and discharge conditions, making the

[2] «A 100% renewable electricity mix? Analyses and optimisations», http://www.ademe.fr/en/a-100-renewable-electricity-mix-analyses-and-optimisations

overall electrical infrastructure more complex. Being able to adapt the consumption to make it as close as possible to the effective renewable production is therefore beneficial even with ESD. The presence of ESD makes it even more important to *plan* how and when the energy will be used. Such schedule allows, by taking into account the capacity, aging and efficiency of the ESD, to decide how to use the available power in order to minimize the ecological impact or the total cost.

1.2 Problem statement and research challenges

A lot of researches are currently done on RES and ESD integration in datacenters [105] and several research projects investigates this area. The ANR project Datazero[3] [152], started in 2015, targets medium scale datacenters (up to 1 MW peak consumption) powered solely with on-site RES, batteries and fuel cells. By mixing researchers in computer science, working on datacenter energy management, together with researchers in electrical engineering, it aims to address finely both sides of the overall problem. However, contrary to existing works, the project looks at optimizing computing resources and electrical resources separately, using a negotiation process between them to find a compromise.

This thesis is focused on a similar scope, with some contributions being related to the research questions explored by the Datazero project. A single datacenter is considered, powered by a rich electrical infrastructure potentially including ESD along with intermittent and uncertain RES. The underlying management problem consists in optimizing the functional behavior of the datacenter itself (*i.e.* running the workload accordingly the user expectations) while minimizing the ecological impact caused by its energy consumption. Instead of trying to solve the overall monolithic problem, it is possible to address it through the coordination of two partially independent parts, in charge of electrical and computing infrastructure respectively, as it is proposed in the Datazero project.

The primary goal of this thesis is to investigate how to handle intermittent RES in such coordination problem, in a way that is as much agnostic as possible of the underlying infrastructures and objectives of each part. A secondary goal is to study how the unavoidable uncertainties of RES power production affect the overall optimization problem and to find methods to take them into account. These questions are addressed by different contributions detailed along this manuscript.

1.3 Main contributions

Analysis and synthetic generation of large-scale workload

In order to perform any optimization on a datacenter management, characterizing its the workload is a crucial step. Several traces of real datacenter workloads are publicly available. Some scheduling approaches in the literature pick one of them for their performance evaluation depending on their models, the scale and the kind of datacenter they target. Others use private data sources, or generate a workload using custom characteristics. This state makes the comparison of approaches difficult, even with similar workload models, as their results are very dependent of the considered workload. It also leads to reproducibility issues, or in some cases to a low confidence on the results of approaches evaluated with unavailable data sources. By analyzing one of the largest publicly available trace, from a Google production cluster, we extracted its main characteristics. Based on the identified

[3] ANR Datazero: http://datazero.org

features, a synthetic workload generator is proposed in section 3.2. It is controlled by a few meaningful parameters. Hence it enables scheduling approaches to be evaluated either with a workload similar to the one of this cluster, or with different hypothesis translated in custom but comprehensive parameter values.

Scheduling algorithm for batch tasks with limited knowledge of electrical model

Several scheduling algorithms were designed recently for managing datacenters with on-site RES. It is achieved by including a part of the electrical sources model in the scheduling problem considered by these algorithms. Therefore the resulting approaches are tied to a specific electrical infrastructure. A new scheduling algorithm, detailed in chapter 4, is proposed to handle this problem in a different way. The electrical infrastructure is considered as a black box which gives indications of its preferences using an aggregated and normalized metric. The proposed scheduling algorithm is agnostic of the type of sources, of possible energy storage devices and internal objectives of the electrical part. An experimental study shows that despite the limited information it reaches similar performances compared to an approach from the literature with full knowledge of the electrical model. This opens the way for flexible scheduling approaches, that adapts easily to new electrical infrastructure or to advances in the sources and storage management methods.

Cooperative and multi-objective power planning with black-box models

The management of datacenter with rich electrical infrastructure is often viewed as a scheduling problem. An alternative vision is proposed, where two different decision modules manage respectively the IT and the electrical infrastructure. Each is solving a distinct scheduling problem under constraint with its own objectives, for a given distribution of power in a time window. The IT part aims to schedule the workload and manage the machines in order to keep the quality of service of the datacenter high, while respecting a power consumption envelope. The electrical system looks at minimizing either the overall cost or environmental impact for providing the given power over time, by scheduling when and how to use each source or energy storage device. Hence the overall problem becomes to find power envelopes leading to acceptable trade-offs between the different objectives. The problem is studied from a purely multi-objective point of view, by trying to find the set of Pareto-optimal trade-offs. Multi-objective evolutionary algorithms (MOEA) are usually well suited for this kind of problem, with the power envelope being the possible solutions and each decision module being used as an objective function. However, due to the costly nature of the associated scheduling problems and the limited time to take a decision, a classical MOEA would lead to poor results. To tackle this issue, a new surrogate method is proposed together with an adapted variant of an existing MOEA. Both are evaluated using a simplified linear model and a realistic IT model with a task scheduling algorithm from the literature.

Optimal policies for datacenter management with uncertain renewable sources

The uncertainty of solar and wind energy production forecasts is a known problem for planning the usage of the energy in a future time window. In datacenter management, inaccurate forecast can affect the quality of a decision or even make impossible to implement a solution if no electrical grid connection is available as a rescue power supply. This

uncertainty is intrinsically tied to the quality of weather forecasting, hence likely impossible to fully avoid. We propose therefore to study a model in which probabilistic aspect of RES production is fully accounted for. A datacenter with RES and ESD as the only electrical sources is considered, using simplified workload and machine model to focus on the uncertain power production aspect. A Markov Decision Process (MDP) formalization of several variants of the problem is used to provide a detailed analysis of their optimal policies and exhibiting their structure. By leveraging these structural properties, a method for computing a bounded approximation of an optimal policy is described and evaluated.

1.4 Plan of the manuscript

The rest of this manuscript is organized as follows.

Chapter 2 starts by introducing some background in optimization methods, datacenter energy management and electrical sources models. The related works are then detailed, along with a taxonomy of existing approaches.

Chapter 3 presents the challenges related to simulation environment for evaluating renewable-aware datacenter management approaches. Existing public data sources for both RES power production and workload traces are discussed. Finally, an analysis of large scale workload traces from a Google cluster is presented along with a synthetic generation method of traces with similar characteristics.

Chapter 4 presents a scheduling algorithm for batch tasks agnostic of the electrical infrastructure. It is evaluated and compared to an existing renewable-aware scheduling approach from the literature with full knowledge of electrical model.

Chapter 5 presents a multi-objective optimization method for finding power planning trade-offs between independent datacenter and an electrical decision modules.

Chapter 6 details a study of datacenter workload management with uncertain RES power forecast. Using MDP framework, an analysis of structural properties of the problem is conducted leading to efficient methods for finding optimal policies.

Chapter 7 finally concludes this thesis and proposes perspectives.

Chapter 2

Background and related works

The three rules of the Librarians of Time and Space are: 1) Silence; 2) Books must be returned no later than the last date shown; and 3) Do not interfere with the nature of causality.

— Terry Pratchett, *Guards! Guards!*

Contents

The works presented in this thesis explore different aspects of the interactions between a datacenter and a rich electrical infrastructure, including renewable energy sources (RES) and energy storage devices (ESD). Figure 2.1 illustrates the different components of such setup. An ideal management should optimize the ecological impact, the operating cost, revenues and the user satisfaction at the same time. To do so, it may act by means of many leverages, from the operating state of the ESD to machine-level energy management mechanisms and workload scheduling. It also requires to take into account the characteristics of the electrical sources, ESD, machines and workload, along with the many uncertainty sources such as the power sources production or user activities. While no work achieves, or even targets this perfect optimization, it gives a good overview of the challenges of this research area.

This chapter provides some background on multi-objective optimization and Markov decision processes, introducing tools and notations used in the related works and along the next chapters. Models related to both energy-efficient datacenter management and electrical sources are then presented. Then, a detailed review of the literature on datacenters management with RES is presented, followed by a conclusion which highlight the positioning of this thesis.

2.1 Mathematical background and considerations

This section aims to present two areas of optimization theory which are heavily used along the works developed in this thesis. Multi-objective optimization is at the core of the

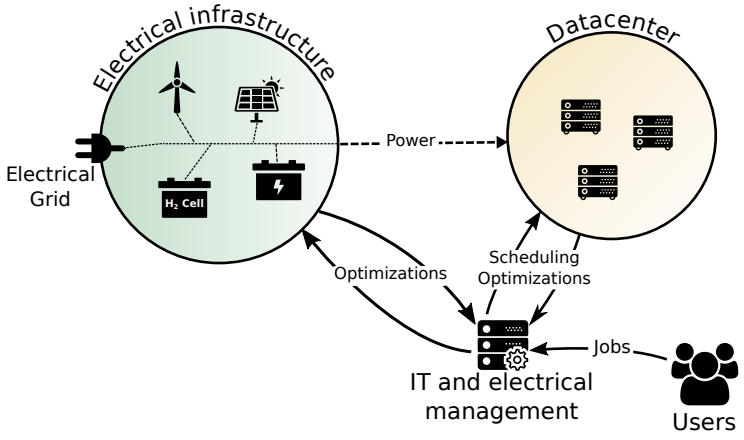


Figure 2.1 – Infrastructure of datacenter powered with renewable energy sources, energy storage devices and electrical grid.

approach detailed in chapter 5 and some concepts are also used along chapter 4 and in most existing works, even when it is not explicitly highlighted by the authors. Chapter 6 is focused on tackling uncertainties and is entirely built on Markov Decision Process (MDP) theory.

The following presentation and discussion about these research areas do not attempt to be exhaustive or to substitute to the rich existing literature. It is instead a summary oriented towards the needs of this thesis, with some references for a reader interested to dig deeper into these fields.

2.1.1 Multi-objective optimization

In many real-life optimization problems, there are several criteria to minimize or maximize simultaneously, some being conflicting with the others. Buying a product is an everyday example of multi-objective optimization problem. When considering a new laptop, a buyer generally wants to minimize the price while maximizing the performances and battery life. It is likely that the cheapest laptop will not have the best hardware for performances. Similarly, high performance hardware is expected to be more power-hungry, leading to lower battery life. Hence, the usual situation in this example is that no product is optimum for all the criteria and the buyer needs to consider the different options *subjectively* to decide a *good* trade-off to choose.

The theory of multi-objective optimization is about formalizing what are the *good* trade-offs in such problem and providing methods to implement the *subjective* choice among them. Using a simplified notation of Deb [45], maximization problems with M objectives can be generically written as:

$$\begin{aligned} &\text{Maximize} && f_m(x), && \forall m \in \{1, \dots, M\} && (2.1)\\ &\text{subject to} && g_j(x) \geq 0, && \forall j \in \{1, \dots, J\} && (2.2) \end{aligned}$$

A value of $x \in \mathbb{R}^n$ satisfying all the constraints g_j is a *feasible solution* of the problem. The set of all the feasible solutions is named the solution space $\mathcal{S} \subseteq \mathbb{R}^n$. Each of the M objectives is characterized by an arbitrary objective function $f_m : \mathcal{S} \to \mathbb{R}$. Hence any feasible solution x can be associated to its *objective vector* $f(x) = (f_1(x), \cdots, f_M(x))$. The space of the objective vectors is named *objective space* here, denoted $\mathcal{Z} \subseteq \mathbb{R}^M$.

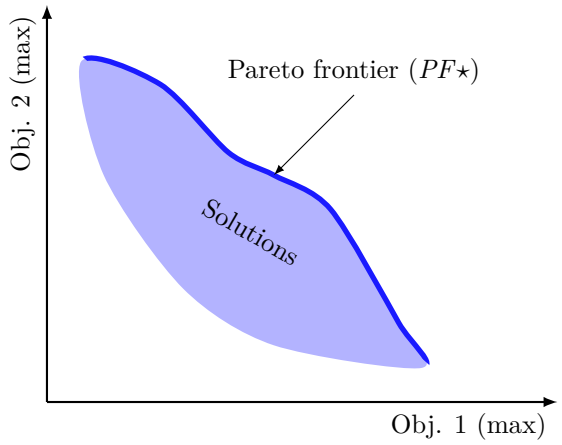


Figure 2.2 – Pareto frontier in a continuous bi-objective maximization problem.

All the ambiguity of multi-objective optimization lies in the ordering of the objective space $\mathcal{Z}$. Without more information on the problem, total ordering is impossible. A partial ordering $\prec_{\mathcal{Z}}$ is however easily defined. Let two objective vectors $z, z' \in \mathcal{Z}$. We say that z is dominated by z', denoted $z \prec z'$ if and only if:

$$\forall m \in \{1, \ldots, M\}, \quad z'_m \geq z_m \tag{2.3}$$
$$\exists m \in \{1, \ldots, M\}, \quad z'_m > z_m \tag{2.4}$$

This ordering may be interpreted as the fact that a solution x' may be better than another solution x on at least one objective, while being at least as good as all the others: $f(x) \prec f(x')$. It is therefore possible to claim that x is clearly inferior to x'. The ordering being partial, some sets of solutions lead to objective vectors that are not dominated by any others, *i.e.* $f(x) \not\prec f(x') \wedge f(x') \not\prec f(x)$. The set of all existing non-dominated solutions form the Pareto frontier of the problem, denoted $PF\star$ and also referred as the *true Pareto front* later. It is illustrated in the fig. 2.2 for a continuous Pareto front.

When a solution must be chosen in a multi-objective optimization problem, picking one from the Pareto front is preferable. The decision on which one to choose, *i.e.* which of the non-dominated trade-off is *subjectively* the best for the decision maker, requires to *articulate its preferences* about each objective. Existing methods are often classified in *a priori* articulation (before to solve the problem) and *a posteriori* articulation (after some or all the trade-offs are known) [137]. For sake of completeness, it should be noted that a few other methods aim to select a solution without articulating the preferences [137] but are not discussed here.

A priori methods are often used to solve multi-objective problem. The main reason is that some of them involves transforming the multi-objective problem into a single-objective one. This avoids the complexity of finding the Pareto front of the problem and additionally allows to use many traditional heuristics designed for single-objective problems. Several a priori methods are commonly used, which are extensively detailed and commented by Ehrgott [60]. One of them is the linearization of the objective vector using a weighted sum, such as the objective becomes:

$$\text{Maximize} \quad \sum_{m=1}^{M} \lambda_m f_m(x) \tag{2.5}$$

With $\lambda_m \geq 0$ being the weight associated to the m^{th} objective. While very common and leading to a solution on the Pareto front, this method has many deficiencies [136] such as

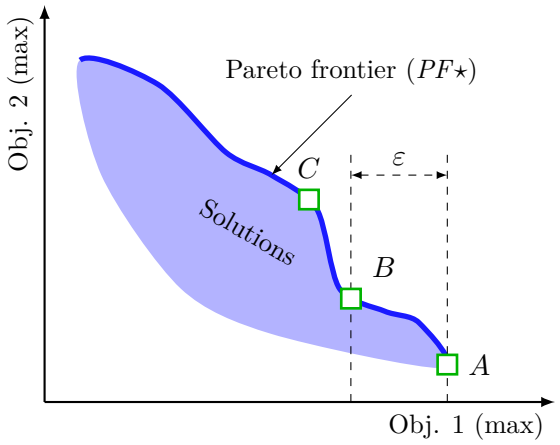


Figure 2.3 – Bi-objective maximization problem showing potential limitations of ε-constraint method. In this case, solution B would be chosen, without even considering the solution C.

the difficulty to predict the impact of changing slightly the weights and the impossibilities to find some non-dominated solutions if the Pareto front is not convex.

Another method referred as ε-constraint, consists in transforming all but one of the objectives into constraints. The optimization problem become, by denoting k the objective kept:

$$\text{Maximize} \quad f_k(x) \tag{2.6}$$

$$\text{subject to} \quad f_m(x) \geq \varepsilon_m, \qquad \forall m \in \{1, \ldots, M\} \setminus k \tag{2.7}$$

The ε-constraint method also gives a solution on the Pareto front. Depending on the problem under study, deciding of the constraints values may be easier than giving weight to each objective, as each individual ε_m value is independent, thus being more meaningful for a decision maker. The main issue is that for some instances of a problem, some constraint values may lead to an infeasible problem, hence no solution. Having such hard constraints with only *a priori* view of the problem may also cause some solutions violating slightly a constraint to be ignored while a decision maker would have considered them otherwise, as shown by the fig. 2.3.

Articulating the preferences a posteriori in multi-objective problems gives to the decision maker an overall view of the possible trade-offs for a given instance of the problem. The first step is therefore to find some or all the solutions of the Pareto front. Finding the true Pareto front is however not always feasible, as in the general case it involves to compute the objective vector for every value of the feasible solution space $\mathcal{S}$. Finding an approximation of the true Pareto front of a problem may be enough in many real-world problems. The research on evolutionary algorithms, a well known class of metaheuristics commonly used for single-objective, led to many promising approaches for multi-objective problems.

Multi-objective evolutionary algorithms

The base principle of evolutionary algorithms is to manage a set of possible solutions using randomness and bias toward the reinforcement of the most promising solutions according to the objective to be minimize or maximize. Each solution is encoded in a form suitable for the considered algorithm, also referred as *individual*. The set of individuals existing in

the algorithm at a given time, or the so-called *population*, slowly changes to explore the solution space (randomness) and try to converge toward an optimum. While the details change from a specific algorithm to another, they usually repeat a simple pattern until some termination criterion is reached. Each iteration, known as a *generation*, creates new individuals by introducing random changes in some existing ones (mutation) and merging the encoded solution of some of promising individuals (mating). Then, a *selection* step is performed to keep good individuals for the next generation. This selection uses the *fitness* associated to each individual, which is generally the value of the objective function for the represented solution.

The field of multi-objective evolutionary algorithms (MOEA) aims to provide such metaheuristic suited for finding approximations of Pareto fronts. Because they handle multiple solutions simultaneously, as a population, they are naturally able to output a *set* of solutions in a single run. This research field, which is still very active, produced many algorithms often bio-inspired and designed for this purpose [201]. While the bases are the same as traditional, single-objective evolutionary algorithms, each MOEA has some specificities for handling the multi-objective aspect. The most important change is often the way the fitness is computed. As the goal is to approximate the Pareto front, most MOEA such as NSGA-II [46] and SPEA2 [207] use the concept of non-dominated solution to compute the fitness of an individual.

Black-box functions and surrogate methods Usually, an optimization problem concerns a well defined model. In some situation however, it may be desirable to optimize the output of a *black-box* function, for which the explicit model is not known but some minimal properties are expected to hold (such as linearity or continuity). This may be the case, for instance, of an objective based on a large n-body or event-based simulation. In multi-objective problems, MOEA are especially well suited for finding an approximation of the Pareto front with black-box objective functions. Evolutionary algorithms in general are based on the creation and evaluation of large number (often thousands or millions) of potential solution. If the black-box objective functions are computationally expensive, this leads to a huge amount of time to reach an acceptable quality of solution.

A strategy to handle this issue is to use *surrogate models* of the real objective functions [22, 93], which are cheap approximations of the target functions using linear or non-linear regression methods. It is possible, by knowing the problem and the instance to solve, to create a surrogate offline and to simply substitute it to the real objectives, but MOEA provide an interesting base to create the surrogate model *online*. Several works attest the efficiency of surrogate-based MOEA [3, 19, 98, 138], but often for a very specific problem. The existing works for integrating MOEA with online surrogate models are discussed more deeply in chapter 5, in which a novel approach is proposed and evaluated.

Performance evaluation metrics for Pareto front approximation

Measuring performance of multi-objective heuristics is, by itself, a whole research area. The major desired properties of a solution set are easily described informally: ideally the solutions should be close to the real Pareto front (*accuracy*), widely and well distributed along the front (*diversity*) and the set should contain as many solutions as possible (*cardinality*) [146, 164]. Defining metrics to measure these properties is however challenging, leading to dozen of different performance metrics [164]. Among the existing ones, two of them, measuring the accuracy and the diversity, are used to evaluate the approach detailed in chapter 5.

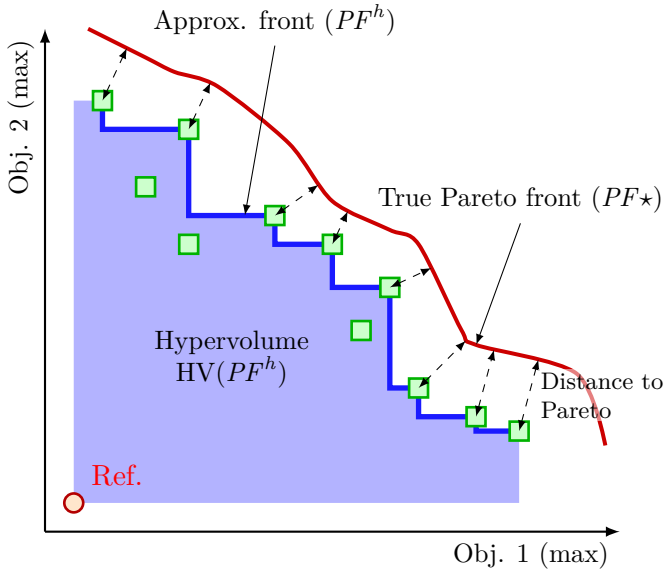


Figure 2.4 – Illustration of hypervolume and distance to Pareto front metrics with 2 objectives. Each square is a single trade-off from a solution set.

Hypervolume The hypervolume indicator (HV) introduced in [205], measures the space covered between the solution set and a pessimistic reference point in the objective space. It is illustrated by the light green area in fig. 2.4. Its value depends on both diversity (how well distributed the solutions are along the Pareto front) and accuracy (how close each solution is from the real front). It may be considered as a global quality indicator without providing alone finer details. In order to compare the hypervolume between different experiments, the same reference point must be used.

When the true Pareto front, denoted $PF\star$, is available, it is possible to use it to normalize the hypervolume, making their values easier to interpret. The normalized hypervolume of a solution set PF^h is given by eq. (2.8).

$$nHV(PF^h) = HV(PF^h)/HV(PF\star) \tag{2.8}$$

Generational Distance Initially proposed by Van Veldhuizen and Lamont [190], the generational distance (GD) measures the average euclidean distance between each solution and the nearest member of the true Pareto front. In fig. 2.4, it corresponds to the average of the dashed arrows. As the complete Pareto front is often impossible to obtain, the best known approximation may be used. For a solution set PF^h, generational distance is given by eq. (2.9)

$$GD(PF^h) = \text{avg}_{u \in PF^h} \left(\min_{v \in PF\star} \left(dist(u, v)\right)\right) \tag{2.9}$$

This metric measures only the accuracy aspect. But hypervolume and generational distance together provide therefore a good overview of the overall quality on both accuracy and diversity of the solution. Because of its name being slightly misleading, in the rest of this thesis the generational distance is also referred as *average distance to Pareto front*.

2.1.2 Decision making and optimization under uncertainties

Uncertainties are more the norm than the exception in real-world problems. Even for deterministic processes, it is rarely feasible to obtain all the knowledge required to model

them exactly. For instance, the number of users accessing to a website at a given time in the future is highly unlikely to be predicted correctly. However, based on a limited amount of knowledge (*e.g.* historical data, correlations with other events), an imperfect prediction model can be available. When such an uncertain information is used in an optimization process, it is common to simply ignore this aspect and to model it as if it was deterministic. Depending the problem, this may be justified because uncertainties are expected to have a negligible impact or because it is acknowledged that the optimization is done with such simplifications.

Uncertainties can nonetheless be accounted in optimization problems, based on additional information about the uncertain data. Several methods can be used to model the uncertainty [104], from a lower and upper bounds of the possible values to a complete probability distribution of the possible realizations. The latter is often more difficult to obtain, as it requires deeper understanding of the random process causing the uncertainties. More complex uncertainty models can be used if the random variables are not independent from each others or if they represents some temporal evolution of a stochastic process [104].

A wide variety of techniques are designed to handle such uncertain problem, generally at the cost of some additional complexity compared to deterministic optimization [109, 122, 167]. Many approaches are based on a notion of *scenarios*, which are either random samples from the known distribution of uncertain variables (Monte Carlo simulation) or possible values chosen specifically (*i.e.* worst case). Each of these scenarios can therefore be treated as multiple deterministic versions of the problem. It is possible, for instance, to construct a linear programming version of the stochastic problem by aggregating the different scenarios, sharing the same decision variables. By setting appropriately the objective (*e.g.* minimizing the average deterministic objective of each scenario), a solution taking into account different outcomes can be chosen [55]. Alternatively, evolutionary algorithms can be used by relying on scenario-based evaluation of the objectives [123, 145]. MOEA present an additional interest here, even when the underlying deterministic problem has a single objective. An additional objective can be created, representing for instance the probability of a given solution to be feasible according to the uncertain variables. Then, a MOEA can be used to obtain the set of ideal trade-offs between the main objective and the risk [97].

Sequential decision making is a special case of general optimization problems. A decision (or an action) has to be taken repeatedly along a time horizon (which may be infinite). Each decision may lead to a short-term reward and an optional long-term reward can be additionally granted at the end of the horizon considered. When uncertainty is considered in this kind of problems, generic approaches presented above can be used, but additional methods are applicable. Particularly, Markov decision processes (MDP) can model many uncertain sequential decision making problems and have been extensively studied for several decades.

Markov decision processes

Markov Decision Processes, or MDP, is a formalization of some sequential decision making problems with uncertainty. We focus here only on finite horizon MDP, with the considered time horizon denoted $N \in \mathcal{N}^+$. At each time $t \in \{1, \dots, N-1\}$ (also called decision epoch), the decision maker must choose an action, denoted a_t, among a set of possibilities $\mathcal{A}_t$. The system, represented by its state $s_t \in S$, evolves depending on the action chosen at the previous epoch and on stochastic part.

Depending the current state and the action taken, a potentially stochastic reward is obtained, denoted $r_t : S \times \mathcal{A} \to \mathbb{R}$. After the last decision epoch, a reward can be

additionally granted depending solely on the final state $r_N : S \to \mathbb{R}$. The evolution of the states is described by the probabilistic transition function $p_t : S \times S \times \mathcal{A} \to \mathbb{R}$. We denote $p_t\left(s'|s,a\right)$ the probability, at the epoch t, of ending in state s' by doing action a in the state s.

A policy is a function which attribute, at each time, an action to do depending on the current state. We denote $\pi = (\pi_1, \ldots, \pi_{N-1})$ a policy for the whole horizon, with $\pi_t : S \to \mathcal{A}$ giving the action to chose at the epoch t for each possible state of the system. The total expected reward can be written recursively, by considering $V_t^\pi : S \to \mathbb{R}$ to be the accumulated expected reward at a given state, from epoch t to the end of the horizon N when the policy π is used.

$$V_N^\pi(s) = r_N(s) \tag{2.10}$$

$$V_t^\pi(s) = r_t(s, \pi_t(s)) + \sum_{s' \in S} q_t\left(s'|s, \pi_t(s)\right) \cdot V_{t+1}^\pi(s') \tag{2.11}$$

The MDP optimization problem consists therefore in finding a policy, called optimal policy $\pi^\star$, such as it minimizes or maximizes $V_1(s)$ for all or for a given $s \in S$. Several methods exists to find such an optimal policy or a good approximation. When the state space S is discrete, backward induction (dynamic programming), policy iteration or linear programming can be used to find the exact solution [154, 155]. All the exact solving algorithms for general, discrete MDP case, are sensitive to the size of the state space. They become costly for large number of states and often not practicable for many-dimensionalty states (curse of dimensionalty) [112, 154]. MDP with continuous state and action spaces cannot be solved exactly in the general case. They can be approximated by discretization of states and transitions probabilities, but doing so requires to increase the state space as much as the desired accuracy of the approximation.

Using the MDP framework can nevertheless be more beneficial by analyzing the resulting formulation. Some MDP can be solved more efficiently by finding structural properties of their optimal policies [84, 126, 154, 155]. For instance, monotonic policies state that the optimal action is non-decreasing or non-increasing with a given ordering of states and actions, resulting in an opening for faster algorithms by reducing significantly the search space [84, 154]. Some of the simpler MDP can even admit an optimal policy with a closed form [155], but any non-trivial problem is unlikely to be reduced to such a simple form. Many other properties can be leveraged to design fast algorithms for exact or bounded approximation of MDP optimal policies, helping to address continuous or many-dimensional problems. Contrary to the generic MDP solving methods, a detailed analysis is required to identify the structure of a specific problem.

Finding an efficient algorithm based on MDP formulation presents many advantages for taking uncertainty into account, compared to other approaches. For instance, scenario-based optimization for uncertain problems can only consider a limited number of possible realizations of the random variables. In addition, the number of scenarios required to probe each uncertainty distribution equally grows exponentially with the number of random variables [112], hence with the number of time steps in sequential decision making problems. At the contrary, a Markov decision process considers the entire distribution of each random variable and are can be solved with a linear complexity depending on the size of the horizon.

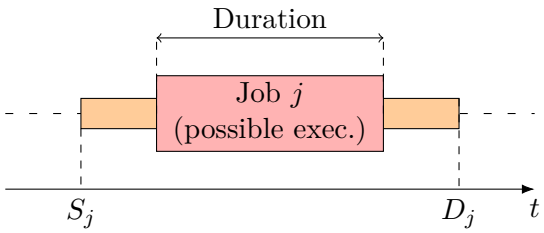


Figure 2.5 – Representation of a batch-like job with its main characteristics.

2.2 Datacenter and power consumption

At its most basic level, a datacenter is a building incorporating a set of servers, networking devices and cooling infrastructure, which goal is to run a workload of some kind. Many kinds of datacenters exist [14], with various scales (from a dozen of servers to thousands of them) and operated with different objectives. Some are operated privately, for a company or institution usage, while others are public, accepting any client able to pay for its computing needs. They can be built for a very specific workload, such as scientific computation, or aiming at covering different use cases depending on the needs of the clients.

Such diversity and large definition makes it rather difficult for a work to address all kinds of datacenters or even all the aspects of a subset of them. Many existing works deal specifically with network optimization in datacenters [16], designing more efficient cooling infrastructure [200] or reducing the cooling cost by taking thermal aspect into consideration [31, 38, 181]. In this thesis, we focus on the servers management, ignoring the networking and cooling aspects.

2.2.1 Workload and quality of service

When looking at datacenter management methods, the considered kind of workload plays a critical role. As mentioned, all datacenters do not aim at running work of similar nature: high performance computing tasks do not share many characteristics with a database server.

A common classification consists in two main types of workload: *batch-like* and *interactive services* [4, 129]. The interactive services are usually designed to continuously, in order to respond to user requests directly (*e.g.* web server or online gaming platform) or indirectly from other services (database). Due to their interactive nature, users expect these services to react as fast as possible, in seconds or milliseconds time scales. Hence their resource usage is closely tied to the amount of requests performed by the users at a given time.

The batch-like jobs are instead defined as a finite and determined amount of work to perform, whose result is expected to be available in a matter of hours or days. While scientific computation perfectly fits in this category, many other common usage are also included such as maintenance tasks (indexing data or integrity checks), video encoding, big data analysis... Such jobs are somehow delay-tolerant, but this tolerance may vary accordingly to the user expectations. It may be a strong constraint, in which case a *deadline* is specified after which the job is considered as failed and the execution stops. If it is a soft constraint, the *due date* denotes a time after which a user may claim to be unsatisfied. A batch job j is characterized at least by a *submission time* S_j, a set of *resource requirements* (denoted r_j^i with i the type of resource), a *duration* or a way to

estimate it (such as a quantity of work) and the *due date* or *deadline*, both denoted D_j. Figure 2.5 illustrates a possible execution of such batch job j respecting its deadline.

For any kind of workload running in a datacenter, it is possible to define some criteria of quality to describe how well it is running according to user expectations. Many different criteria are identified in the literature, which relate to performances, security, availability or reliability aspects [80, 108]. The user expectations are theoretically defined in a Service-Level Agreement (SLA), which determines a set of criteria and associated metrics along with accepted level. In practice, large operator often impose predefined SLA to small clients which are not important enough to negotiate the conditions in detail [87]. Such a metric used in the definition of SLA is called a Quality of Service (QoS) metric in the following. For instance, the availability of a service may be measured as a percentage of time it is available over a year and a SLA may specify that this QoS metric must be greater than 99.5%.

Even when focusing only on the performance aspect, there are multiple QoS metrics used in the literature. For services, the response time to a request is commonly used [199]. In shared environments, such as in virtualized infrastructures, the availability of requested resources such as CPU cycles is also widely used in the literature [203]. For batch jobs, the most common QoS metric used is the respect of the deadline or due date [27, 127].

2.2.2 Server power consumption

Modeling the power consumption of machines has been the subject of many studies. The existing models, well referred and classified by Dayarathna et al. [43], are targeting different trade-offs between the complexity (or available information) and the resulting accuracy.

Basically, the power consumption of a powered-on machine is divided in two parts: a static power consumption denoted P^{idle} and the dynamic power consumption P^{dyn}.

$$P^{total} = P^{idle} + P^{dyn} \tag{2.12}$$

The static part includes the power of all the components in their lowest activity state (disks idling, motherboard components, processor with minimal activity, fans, network card). The dynamic part may be again split into the additional power consumed by each component in a higher usage state. Without loss of generality, it can be written as a function $\text{dyn} : \mathcal{S} \to \mathbb{R}$ of the overall state $\mathcal{S}$. This function can in turn be detailed for each component, depending on the desired granularity and possible focus to some specific parts, as illustrated by eq. (2.13).

$$\text{dyn}(\mathcal{S}) = \text{dyn}_{cpu}(\mathcal{S}_{cpu}) + \text{dyn}_{disks}(\mathcal{S}_{disks}) \cdots + \text{dyn}_{ram}(\mathcal{S}_{ram}) \tag{2.13}$$

While some works attempt to model very finely the power consumption [39], simple models prove to be quite accurate. The simple linear model in eq. (2.14) proposed by Fan et al. [64] in 2007, approximate the overall dynamic consumption at the server level based only on the CPU usage u. Even if it is both dated and over-simplified, the average error on this model has been measured to be lower than 6% with more modern hardware [39]. Hence it is used in many recent works, when a finer model of the hardware is not desired.

$$P^{total} = P^{idle} + u\left(P^{max} - P^{idle}\right) \tag{2.14}$$

A typical situation where a more detailed model is required is when an approach uses Dynamic Power Management (DPM) techniques to act on the power consumption [12]. Dynamic Voltage and Frequency Scaling (DVFS) focused a lot of attention during the last

decade. Modern CPUs support several combinations of frequency and voltage, which may be selected depending the desired trade-off between processor performances and power consumption. The relationship between frequency and voltage is usually managed automatically, such as the lower stable voltage is set for any given frequency. Hence, the classical models for DVFS only consider the frequency, denoted f_n for a given processor n. Only certain frequencies are allowed, such as $f_n \in \mathcal{F}_n$ with $\mathcal{F}_n$ the set of possible values for this processor. The performances are generally assumed to scale linearly with the frequency, while the power consumption follows a different law. A simple model, still used in recent works [27], extends the previously detailed linear model to account of each processor power, is given in eq. (2.15) for a machine with N processors. It considers that the base dynamic power of a processor depends on the cube of the frequency. The term P^{dyn} is a processor characteristic and u_p denote the processor usage ratio.

$$P^{total} = P^{idle} + \sum_{n=1}^{N} u_n \cdot P_n^{dyn} \cdot (f_n)^3 \tag{2.15}$$

In order to get rid of the idle power of a server, it is possible to act on the global *power state* of a machine, as described by the Advanced Configuration and Power Interface (ACPI). Many works consider the possibility to switch on and off machines [13, 192], or to put them into sleep modes (suspend to RAM or suspend to disk) [71]. However, powering a server on and off is not instantaneous and consumes extra energy. The power consumption during the early boot process may even exceed the peaks of power monitored during high load activity [147]. The boot and shutdown times depend on many factors, such as the operating system, its storage medium (spinning hard drive or SSD), the software configuration and the services expected to be available to consider the machine ready [179].

Modeling the power state management requires to define the considered states (such as on/off/idle) along with the transitions between them. Each state s is characterized by a power model P^s, such as eq. (2.12) for the *on* state. The off or idle power consumption is usually modeled by a machine-specific constant. This value may be 0, for instance when a power off machine is also considered to be disconnected from the power circuit by a manageable Power Distribution Unit (PDU). Each possible transition between a state s to another state s' is characterized by a transition time $t^{s \to s'}$ and an average power $P^{s \to s'}$ consumed during this time.

The different transitions and corresponding power consumptions are shown in fig. 2.6, for a model with two states (`on` and `off`). It also illustrates the power consumption in active state for a linear load model as given in eq. (2.14).

2.2.3 Energy-aware datacenter management

As it impacts directly the exploitation cost, many current researches on datacenter management aim at reducing their energy consumption, without dealing with renewable energy specifically. This section aims to provide a quick overview of these approaches, in order to highlight the similarities and differences with renewable-aware approaches later.

One of the common way to reduce the energy consumption is to make use of DVFS to act on the performance/consumption trade-off. Wang et al. [195] propose several heuristics for scheduling parallel batch tasks with precedence constraints, using DVFS to reduce consumption generated by non-critical tasks, without increasing the total execution time of the set of tasks. In addition, they present a mechanism based on an SLA established with the customer, allowing to reduce the energy needed even more in exchange for a bounded increase of execution time. Von Laszewski et al. [193] present a virtual machine

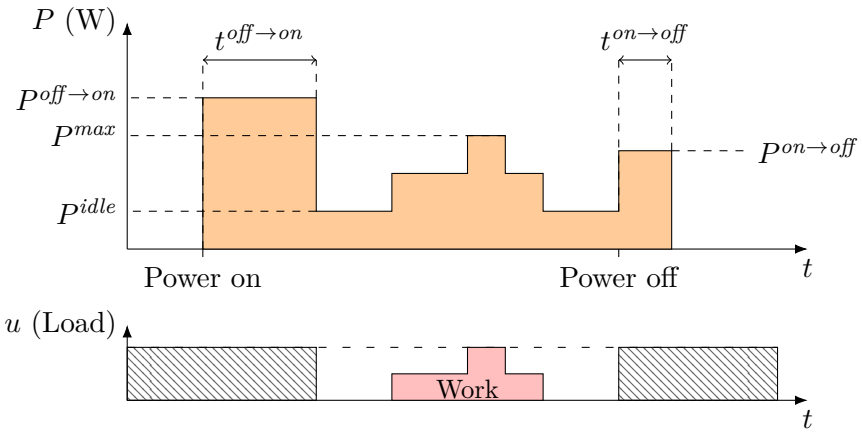


Figure 2.6 – Illustration of power state model (on/off) and a linear power model based on CPU load.

(VM) scheduler for homogeneous, cloud-oriented clusters. They use DVFS to reduce the power consumption of the physical machines, depending on the QoS requirements of the virtual machines running on them.

With the increased use of virtualization in datacenters, some recent approaches leverage such environment to reduce the energy consumption. Zhou et al. [203] present a greedy heuristic for minimizing the energy consumption using consolidation while considering the quality of service of the VM. The QoS metrics consist in resource requirements of VM and in performance degradation caused by migration during consolidation. By measuring the load of each machine, the algorithm try to keep it bounded to avoid both over-loading to preserve VM resources and under-loading to reduce power consumption.

Energy- and thermal-aware scheduling algorithms are often studied in the context of datacenters. Sun et al. [181] give a spatio-temporal thermal model for machines in datacenters. They use this model to develop an online scheduler taking thermal constraints into account, using DVFS to dynamically change the power and thermal impact of the machines. In [117], authors propose several greedy algorithms to improve the cooling infrastructure efficiency and to take into account the high failure rate of large datacenters.

Some works do not try to minimize the energy consumption but the *energy cost*. Li et al. [121] present a scheduling approach to minimize the cost of the energy consumed while guaranteeing security-related QoS. The security guarantee is obtained by adding security services at runtime according to the level of SLA of each task. The scheduling algorithm itself is an online heuristic, based on Lyapounov optimization. With only knowledge of the past and current electricity prices, the algorithm takes the decision of whether executing a task or waiting for cheaper electricity.

Renewable-aware approaches differ from most of the energy-efficient schedulers as the first goal is not to reduce the total energy consumed, but generally the *non-renewable* energy consumed. Therefore, they have more similarities with approaches such as [121], which tries to minimize the energy cost while having dynamic electricity pricing.

2.3 Electrical sources and storage

The traditional works for energy-efficient datacenter often consider the power supply infrastructure with the simplest possible model: an infinite source of electricity. A few works from the electrical engineering community show the complexity of finer models, which de-

scribe the distribution and conversion of the electricity inside the datacenter itself [149]. This distribution requires multiple transformers and converters, some level of hardware redundancy is needed for reliability, along with Uninterruptible Power Supply (UPS) near each server. A total abstraction of the electrical infrastructure is however reasonable for approaches focused only on workload and machine management for energy efficiency in a datacenter powered solely by the electrical grid.

The case of renewable-aware energy management is different. Trying to leverage on-site RES requires at least some rough model of the renewable sources: a finite power source, with available power varying across time. Batteries or other ESD associated with the RES may offer other optimization perspectives, at the cost of additional complexity of electrical infrastructure. Even a traditional electricity provider may be modeled more finely to take the price or the environmental impact of the energy into account.

When the goal is to reduce the environmental impact of energy consumption, it is required to define and to measure it. Climate change is not the only consequence on the environment of human behavior and many other aspects may be considered [197]. Nevertheless it focuses a lot of attention from the scientific communities, resulting in a lot of studies and data available. A common way to measure it to unify greenhouse gas (GHG) in terms of equivalent kilogram of CO_2, denoted kg CO_2-eq, over a fixed time period (dozens of years), therefore including the other GHG.

This section presents some background on the different electrical sources and several storage devices. While some models of the literature are presented and discussed, their main goal is to highlight some important characteristics thus explaining certain choices in existing works or in the contributions presented in this thesis.

2.3.1 Electrical grid

On-site electrical generation and storage systems can be designed either as grid-connected or stand-alone infrastructures. The first are connected to a traditional electricity provider, or *electrical grid*, for either buying energy or selling excess of production. On the contrary, stand-alone systems are isolated and can only use ESD in a limited way to store such excess, with no possibility to acquire more energy. Kaundinya et al. [99] highlight the specific advantages and disadvantages of both. Basically, grid-connected systems can only be deployed in areas with accessible electrical grid and requires additional costs just for the interfacing with it. On the other hand, the grid provides a virtual infinite storage if it is possible to sell an excess of production and increases reliability compared to stand-alone systems. The works on datacenter with RES almost exclusively focused on grid-connected systems.

The electrical grid can be modeled either as a source only [118] or as a bidirectional power device [78]. In both cases, it is generally considered as a perfect source or sink, *i.e.* with no limit of power or other constraints. Many works on datacenter management, including a few which are not designed for RES, uses a varying price of the grid energy [78, 113]. At a given time t, the price of a unit of grid energy is denoted c_t^{grid}. Hence assuming a discrete time model with time steps of duration δ, the cost for consuming an average power p_t from the grid for each time $t \in \{1, \dots, T\}$ is given by the simple sum:

$$C^{grid} = \sum_{t=1}^{T} p_t \cdot \delta \cdot c_t^{grid}$$

Most approaches rely on grid pricing being known perfectly in advance for some time window at the time of the decision. A few works use a simpler model for varying grid

price, based on on-peak and off-peak tariffs [70]. In this case, two prices, θ_{min} for off-peak and θ_{max} are used depending on the hour of the day. Being a specific case of the aforementioned pricing model, it can be transformed into the corresponding set of c_t^{grid} for all t.

For approaches where selling the excess to the grid is allowed, some adjustment is done such as the selling price is not the same than the buying price at a given time. For instance, Gu et al. [78] uses a parameter $\gamma \in [0, 1]$ to obtain the selling price $c_t^{sell} = \gamma c_t^{grid}$. This lower selling price is explained by the conversion and distribution losses, but also by the fact that the grid operator is not always able to absorb additional power.

Identifying and measuring the GHG emissions caused by electrical grid energy is a complex topic. First, it depends highly on the considered location, as each national grid uses different primary sources mixes [88] which produce different quantities of emissions per unit of energy. Then, accurate modeling should also take into account the time of consumption. Even without considering the wind and solar production of the electricity generation mix, conventional power plant are not always enable depending on their characteristics and the electricity demand. For instance, when demand is high (typically during work hours), coal and other fossil fuel power plants are more used, while nuclear power plants, being very slow to start and stop, provide a higher proportion during low-demand time.

Hence the grid GHG may be modeled similarly to the energy cost [51], with c_t^{grid} being the equivalent CO_2 emissions for an energy unit at a time t. Unfortunately, precise sub-day data is not always available, leading to daily or yearly average used instead. When considering selling renewable excess to the electrical grid, this problem is amplified with the difficulty to determine the net impact of this additional energy on the overall grid operation. Similarly to the economic cost, a parameter $\gamma \geq 1$ can be adjusted to reflect the losses and other grid behaviors which lead the injected energy to compensate only partly the equivalent emissions from current generation mix.

2.3.2 Renewable energy sources

The term renewable energy source, according to all the commonly admitted definitions, includes sources such as photovoltaic panels, wind turbines, hydroelectric, geothermal and biomass power plants. However, as stated previously, the researches targeting renewable energy integration with datacenter almost exclusively focus on the sources with intermittency and unpredictability, which are the roots of the scientific challenges. Hence in this section and in the rest of the thesis, the term RES refers primarily to wind turbine or photovoltaic panels, according to its usage in the datacenter-related literature.

Wind turbines and photovoltaic panels share these two important characteristics: being tied to the weather conditions, their available power over time is intermittent and difficult to predict accurately. Both also exhibit some more or less pronounced seasonality and daily cyclic behaviors [15]. That being said, the models used to determine the output power from the corresponding environmental conditions, are totally different between photovoltaic panels and wind turbines and these environmental factors also follow distinct models.

In most of the literature on datacenter management with on-site RES, they are however modeled indistinctly, as generic sources with variable power [78, 119, 150]. With this model, all the renewable sources can be aggregated. We denote $p^{renew}(t)$ the power available at time t for all the RES. This value is not known from the decision algorithm before the time t is reached.

Let $\hat{p}_t^{renew}(t')$ be the prediction known at time t of the renewable energy available at time $t' > t$. The approaches differ greatly on way the prediction is modeled. It can be

considered to be a perfect prediction, at least during a given time window of duration $T^{forecast}$, in which case $\hat{p}_t^{renew}(t') = p^{renew}(t'), \forall t' \in [t, t + T^{forecast}]$ and is undefined for other values of t'. Alternatively, some approaches consider the imperfect prediction by updating the forecast regularly [70]. In such case, at a given time t_0, the value of the prediction for a future moment $t' > t_0$ (denoted $\hat{p}_{t_0}^{renew}(t')$) may not be the same as the prediction for the same moment at a later time t_1 ($\hat{p}_{t_1}^{renew}(t')$). Many methods can be used to compute the renewable prediction based on weather data, which is discussed in section 2.3.2.

In this thesis, the RES used for all experimental results is photovoltaic energy. Because of the simplistic power-over-time model commonly used in the literature, it is possible to consider wind turbines, other intermittent sources or a mix of several renewable sources instead, without significant changes. However, understanding the internal model of RES is still useful, by giving a sense of the relationship between weather conditions and power production. This helps to understand how the weather forecast uncertainties is translated into production uncertainties.

Photovoltaic module models

Modeling the production for PV panels is still an active research field. A large number of models already exist [72, 103, 135, 202], depending on the cells technology, the required accuracy and the available parameters. Most of the existing works target the electrical engineering community and therefore attempt to model the detailed electrical characteristics of the PV system. Notably, the PV modules are known to have a very specific relationship between the intensity of the load connected to it and the output voltage. This aspect is often covered by the models, as it represents a ground-truth required for fine electrical simulations. When using PV modules for practical application however, a dedicated system is in charge of finding the ideal intensity and tension for maximizing the output power (known as Maximum Power Point Tracking, or MPPT) [56].

When modeling a PV system at a higher level, it is more convenient to deal with output power at the maximal power point. Such a model is described as *empirical* by Goss et al. [72] and focuses on the overall system (PV, MPPT and sometimes converters) at the cost of some accuracy. They are however used by the overwhelming majority, if not all the works on RES integration with datacenters. The most important variable to model the PV production is obviously the amount of solar power received by the panels, or *solar irradiance*. Hence the most basic model is called here the irradiance-linear model and is described by eq. (2.16). It gives the output power p^{pv} of a PV array of area A depending on the measured irradiance E at their angle by taking into account the nominal efficiency of the panels (denoted η) and an optional performance ratio ρ of the whole system (including converters and other losses) [72].

$$p^{pv} = \eta \cdot A \cdot E \cdot \rho \tag{2.16}$$

This simple model does not take two major aspects into account: the effect of the temperature on the module efficiency and the non-constant efficiency at different irradiance levels. The first aspect is sometimes considered in the renewable-aware datacenter management literature. For instance, Kumar et al. [110] are using a PV model which takes into account a "temperature exceedance loss", denoted L. Their model is given by eq. (2.17), which also approximates the normal-plane irradiance (E above) from the easier to obtain vertical irradiance E^v and the angle α between panels and the sun.

$$p^{pv} = (1 - L)\eta \cdot A \cdot \cos(\alpha) E^{v} \tag{2.17}$$

A representative and relatively complete model, yet simple compared to the state of the art used in electrical engineering, is PVFORM [140]. It handles the non-constant efficiency of the panels depending on solar irradiance through an alternative formulation when it is less than $125\,\mathrm{W\,m^{-2}}$. This effects corresponds to the decreased efficiency observed in crystalline silicon cells [135] in such conditions. The parameters used by the model are detailed in table 2.1. The required environmental parameters consist only in the irradiance E and temperature T. It requires only few parameters from the PV modules themselves, which are commonly found in the datasheet of recent products, reducing the need of complex experimental measurements. Equation (2.18) gives the formula used to compute the maximum output power with the PVFORM model.

Parameter	Unit	Meaning
p^{pv}	W	Predicted maximum output power.
P_{m_0}	W	Measured output power at Standard Reporting Condition (SRC).
E	$\mathrm{W\,m^{-2}}$	Solar irradiance, as measured with a pyranometer at the location and angle of the panels.
E_0	$\mathrm{W\,m^{-2}}$	Solar irradiance at SRC.
T	°C	Temperature of the cells.
T_0	°C	Temperature of the cells at SRC.
γ	$\mathrm{°C^{-1}}$	Maximum power correction factor for temperature.

Table 2.1 – Summary of parameters used by the PVFORM model and their meaning.

$$p^{pv} = \begin{cases} \dfrac{E}{E_0} P_{m_0} \left(1 + \gamma(T - T_0)\right) & \text{if } E > 125\,\mathrm{W\,m^{-2}} \\ \dfrac{0.008 \cdot E^2}{E_0} P_{m_0} \left(1 + \gamma(T - T_0)\right) & \text{otherwise} \end{cases} \tag{2.18}$$

Lifecycle cost and greenhouse gaz emissions

The manufacturing, maintenance and disposal of renewable power sources are far from being free of environmental impact. For instance, manufacturing photovoltaic panels is quite energy hungry and involves the extraction and transformation of some raw materials. Consequently, even if they produce more energy in their lifetime than they consume during their production, they are definitively a source of GHG emissions [7, 151].

It must be highlighted that the aging of photovoltaic panels is almost entirely tied to the solar radiations they received and their thermal conditions [133]. Hence it depends mainly on their time spent outside, whether power is actually consumed or not.

These two statements explain the different consideration for economical cost and GHG emissions of renewable sources in the literature. It may be meaningful to take these costs into account, for instance when studying the impact of installing new RES or buying renewable energy from a tier operator. In [51, 78] for instance, economical and ecological costs are added, which may be balanced with the potential gains associated with their usage. On the contrary, if the scope of a work is an already built infrastructure, only maintenance cost would make sense. In the case of photovoltaic panels specifically, it appears that the maintenance cost does not depend on their production or on the decision

made on the usage of this energy. Such studies can therefore arguably ignore the cost of the renewable energy production, considered as null in an energy dispatch optimization process [68, 70].

Weather forecasting for photovoltaic panels

Solar and wind energy production is tied to weather conditions. Therefore, weather forecasting methods are used to predict the future production of these RES. As shown previously, the most important factor in all photovoltaic (PV) production models is the solar irradiance received by the panel. The air temperature also affects slightly the production by impacting the panel temperature, but is secondary and more predictable than solar radiations [81]. The irradiance at a given location and time depends of several parameters, such as the position of the sun in the sky (zenith solar angle), the angle formed between the panels and the sun, as well as several weather-related data. The presence and nature of clouds blocking the direct sunlight is the most obvious, but the atmospheric humidity and other conditions affecting the indirect irradiance may affect the overall irradiance significantly [9]. Various metrics have been proposed to represent the impact of the weather conditions on the irradiance, such as the clear-sky index introduced by Marty and Philipona [139] or the KPV [63] which is directly designed for being used in PV forecasting.

Many methods exist for forecasting irradiance and related weather conditions in general, from statistical estimation using past data to satellite imagery processing combined to numerical weather prediction models [54]. The main challenge for predicting production of a single PV array is the need for high temporal resolution and precise location forecast, with an accuracy of a few dozens or hundreds of meters. Traditional numerical weather models are rarely able to provide this precision [63, 132].

Using numerical weather predictions as a basis for single-site irradiance forecasting causes errors of two different nature: a part is due to their relatively low spatial resolution and the other is caused by the weather model themselves. Methods based on historical data, for instance persistent models or AutoRegressive Moving Average (ARMA) can be used for local predictions over short time horizon ($\leq$ 1 hour) but are highly inaccurate for longer periods [54, 106]. Spatial and temporal resolution of numerical weather prediction models can also be improved using different post-processing, by combining historical statistic models to weather forecast or by interpolation and filters [54, 132].

The irradiance forecasting errors and therefore the prediction uncertainties, are correlated by several authors to the clear-sky index or cloudiness [106, 132]. Particularly, Lorenz et al. [132] show the correlation of their prediction error with both solar zenith angle (less error near sunset or sunrise) and clear-sky index, in a way that very cloudy or very clear conditions give more accurate results than partially-cloudy sky. In the same paper, the authors uses this prediction error to compute the prediction intervals of the expected irradiance.

Several other authors recently started to investigate on probabilistic forecasting methods [9]. Instead of giving a single point prediction for each time of the horizon (deterministic forecast), these methods output either a lower and upper bounds, or a complete probability density function of the irradiance. This area is still young and all authors do not agree currently on the best distribution law to fit the irradiance uncertainty. Some works suggest normal or Laplacian distributions are acceptable models [143], but Beta distribution are used by several authors [11, 109] and more complex joint distributions, such as multivariate normal laws were also proposed [143].

Many methods for predicting irradiance exist, with different trade-offs between accuracy, time horizons and both geographic and temporal resolutions. Hence, best predictions for a precise location are achievable using multi-model approaches, combining their respective strengths [106]. Many sources of errors are identified in the literature and cause these predictions to be uncertain. However, detailed models of uncertainties in the form of confidence intervals or probability density functions are still discussed. The PV models presented in section 2.3.2 can be used with minimal effort to predict probabilistic production by knowing the irradiance (and possibly air temperature) probabilistic forecasts.

2.3.3 Energy storage devices

Electrical energy storage at utility-scale is challenging. Many technologies exist, each with specific advantages and drawbacks [48]. All the storage technologies first transform the electrical energy into another form which is easier to maintain. They can be classified based on this intermediary energy form: mechanical (pumped-storage hydroelectricity, compressed air, flywheel), thermal (molten salt storage, steam accumulator), electrochemical (battery, supercapacitor) or chemical (electrolyzer/fuel cell system, methanation).

Each ESD technology has some specificities, requiring dedicated models for a fine-grained study. However, a coarse model of most of them can be achieved with a few characteristics, which are commonly used to compare technologies:

- round-trip efficiency η, representing the expected energy released for a unit of energy absorbed
- self-discharge rate γ, as many technologies cannot retain the stored energy for a long amount of time
- capacity in terms of stored energy, Q^{max}
- maximum power, potentially different for charging (ρ^{charge}) and discharging ($\rho^{discharge}$)
- lifetime, representing the aging of the device, often expressed in number of charge/discharge cycles

Using these characteristics, a dynamic coarse-model with discrete time (time step δ) is described by eqs. (2.19) to (2.21). The energy stored at time t is denoted by Q_t, and p_t is the charged (positive) or discharged (negative) power. It is common to talk about the State of Charge, which is defined as a percentage of the capacity ($100 \cdot Q_t/Q^{max}$).

$$\forall t \quad \rho^{discharge} \leq p_t \leq \rho^{charge} \tag{2.19}$$

$$0 \leq Q_t \leq Q^{max} \tag{2.20}$$

$$Q_t = Q_{t-1} \cdot (1-\gamma) + \begin{cases} \eta \cdot p_{t-1} \cdot \delta & \text{if } p_t > 0 \\ p_{t-1} \cdot \delta & \text{otherwise} \end{cases} \tag{2.21}$$

This model can be even more simplified in some cases. For instance, some battery technologies have very low self-discharge rate, which may be approximated by $\gamma = 0$ in eq. (2.21). While few existing ESD have near-perfect efficiency, a rough approximation of $\eta = 1$ is sometimes used [51, 110]. Similarly, supercapacitors and certain batteries have high maximum power values and it may be acceptable to ignore constraints from eq. (2.19).

Oppositely, finer models quickly exhibit the specificities of each ESD. For instance, the effect of aging differs between each technology: it can affect the capacity Q^{max}, the maximum charge and discharge rate (ρ^{charge}, $\rho^{discharge}$) or the efficiency η. An other example is the impact of charge and discharge rate, which may act on the efficiency,

on the lifetime, and even the *available* capacity for some time (lead-acid batteries for instance).

Among the various ESD, electrochemical batteries are still a reasonable choice for moderate capacity of energy storage. However, battery is a generic term which encompasses dozen of technologies, with very different characteristics [153]. The two families of mature battery technologies which are currently the most used for RES storage applications are lithium-ion and some of the lead-acid ones. Lithium-ion appears to be more and more interesting, but suffers from an important investment price. On the contrary, even if lead-acid batteries have relatively short lifetime (among other drawbacks), they are still economically attractive because of their low prices [153].

While the works of the renewable-aware datacenter management literature use relatively coarse models (similar or simpler to eqs. (2.19) to (2.21)), we propose to briefly describe a more accurate model of lead-acid battery. It aims at illustrating the complexity of the realistic models, highlighting the commonly accepted simplifications.

Fine charge and discharge models of battery

With the wide range of existing models in the literature [95, 188], choosing one of them requires to understand the pros and cons of each. Different approach families exist, such as (i) electrochemical models which take into account the chemical process occurring inside the batteries, (ii) electrical-circuit models which represent a battery as a traditional electrical circuit, (iii) analytical and stochastic models which use a more abstract representation of the batteries.

While electrochemical models, such as in [66], can reach a very good accuracy, they also require to set dozens of parameters related to the materials used inside the battery or to the size and shape of the electrodes, making difficult to adopt in practice. The equivalent electrical-circuit (EEC) models [28, 157] give an interesting abstraction when batteries are integrated in more complex electrical-circuits. They indeed allow to represent them uniformly, and give directly various electrical measurements (tension, intensity or internal charge). However, they are not as accurate as the electrochemical models [95] and often require to get experimental data in order to set their parameters.

Stochastic models [34, 35, 159] achieve pretty good accuracy, by representing the charge or discharge of the battery with Markov chains. They require few parameters, but the resulting Markov chains tends to be huge (between $6 \cdot 10^6$ and $4.5 \cdot 10^8$ states in their experiments [95]), and depends directly on the capacity of the considered battery.

Finally, the analytical family contains several interesting models. One of the simplest is the Peukert's law, which can be used to approximate the battery lifetime when discharged at a constant rate. However the simplicity of this model prevents it to be usable when the output current varies over time, and results in a moderate or low accuracy [95]. Another analytical model is the Kinetic Battery Model (KiBaM) [134]. It considers the battery charge distributed in two "wells": the available well and the bound well, as shown in fig. 2.7. The first well represents the amount of charge directly available, whereas the second one consists in the charge actually present in the battery, which cannot be used at the moment due to the internal chemistry of the battery though. The charge goes from a well to another at a rate which depends on a parameter k and on the difference between the level of both wells (h_1 and h_2 in the figure). The relative size of both wells is controlled with a parameter c, and the charge in each well is given by $q_1 = c \cdot h_1$ and $q_2 = (1-c) \cdot h_2$. Two differential equations model the evolution between the two wells depending of the charge or discharge current I.

With only two parameters, which can be extracted from most battery datasheets,

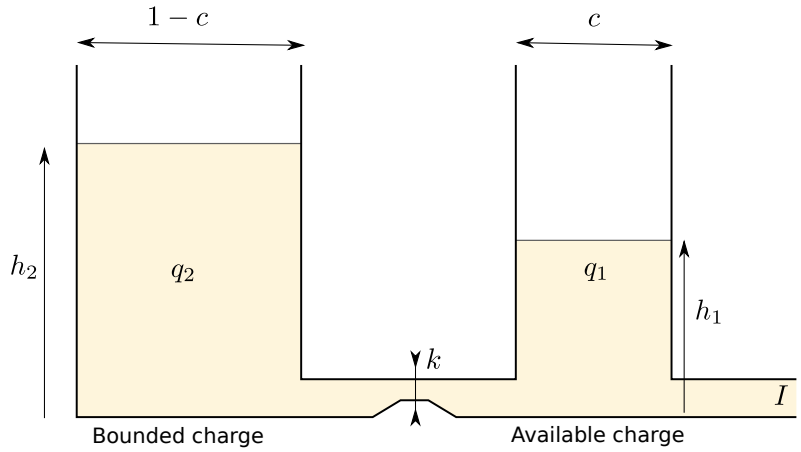


Figure 2.7 – The "two wells" representation of a battery with the Kinetic Battery Model (KiBaM).

the KiBaM model is quite simple. It is designed initially for lead-acid batteries, but is used for other technologies as well, such as lithium-ion [96]. It models quite accurately their behavior with non-constant charging and discharging current. In addition, recent researches improve some lacking aspects of KiBaM, *e.g.* the slight asymmetry between charging and discharging parameters and the impact of battery aging on them [96].

Very few approaches for datacenter management are using any of the previously cited battery model. A notable exception is the approach from Li et al. [115], modeling the behavior of lead-acid batteries through KiBaM equations.

Battery aging and environmental impact

It is rather difficult or impossible to compute the exact amount of pollution emitted to manufacture a complex device, such as batteries. However, some studies on Life-Cycle Analysis (LCA) try to evaluate either the energy needed or the environmental impact of such manufacturing. Depending on the studies, this evaluation may take some steps into account, such as extraction of raw materials, transportation or recycling. This causes the wide range of values found in the different studies.

The impact of batteries is measured in terms of emission per quantity of stored energy, or kg CO_2-eq /kWh, allowing to compare technologies with different energy densities. There is no clear consensus on the GHG emissions for manufacturing batteries, with works finding values from 38 kg CO_2-eq /kWh to 356 kg CO_2-eq /kWh [61, 82] just for lithium-ion ones. It also depends on the exact technology, as lithium-ion encompasses several kind of batteries with very different manufacturing processes. For instance, the same study from Hao et al. [82] found values from 96.6 kg CO_2-eq /kWh for Lithium Manganese Oxide (LMO) to 109.3 kg CO_2-eq /kWh for Lithum FerroPhosphate (LFP), two lithium-ion technologies.

The lifetime of a battery depends on numerous parameters, such as the charge/discharge currents, the ambient temperature and the number and the depth of discharge cycle [79]. However, the number of full charge-discharge cycle is a good aging predictor [33], assuming non-extreme usages (regulated temperature and depth of discharge in the range of the manufacturer). The values commonly used in the literature for the lifetime of lithium-ion batteries are between 1000 and 4500 cycles [33, 59, 186].

Technically, using a battery does not cause additional emissions, as its manufacturing

is already performed. The battery lifetime is nonetheless greatly affected by its usage and it may be considered that once a battery is considered to be too deteriorated it will be replaced by a new one. Hence it is possible to incur some of the initial battery cost (economical or ecological) to each performed operation, depending on how much it affects its lifetime, as a form of amortization. Deng et al. [51] model the cost of using the battery as if any amount of charged or discharged energy during a time step was a full cycle. It leads to a fixed cost of C^{bat}/N^{cycle} for every battery operation, with C^{bat} the total cost of new batteries and N^{cycle} their lifetime in number of cycles.

However, a small amount of charged or discharged energy does not cause the same degradation to the battery than a full charge/discharge [33]. To take this fact into account, a finer model may be used. The impact on aging can be associated linearly with the amount of charged/discharged energy and symmetrically over the charge and the discharge. It gives a cost per kW h of charged/discharged energy, denoted c^{batuse} in eq. (2.22).

$$c^{batuse} = \frac{C^{bat}}{2 \cdot N^{cycle} \cdot Q^{max}} \tag{2.22}$$

2.3.4 Electrical infrastructure management under uncertainties

The integration of intermittent and uncertain RES causes new challenges for controlling and stabilizing electrical grids. This leads, at least partly, to a major change of the grid management paradigm with the recent emergence of the smart grids research area [175]. Works on smart grids encompass many topics which are distant from the primary focus of this manuscript and therefore we do not attempt to provide a comprehensive survey of the whole field here. A few works are nonetheless closely related to the problems studied in the next chapters. In particular, uncertainty of RES attracted attention in the smart grid community recently [49, 109, 122] with approaches comparable in some aspects with the contribution proposed in chapter 6. For this reason, a brief overview of RES uncertainty management is proposed.

Liang and Zhuang [122] review existing works for dealing with uncertainties of RES production and electrical load in microgrids, which are define as partially independent electrical infrastructures with multiple consumers, producers and storage devices. The diversity of existing methods to deal with uncertainties, presented in section 2.1.2, is well represented in these works. Scenario-based approaches are among the most commonly used, but other methods such as Lyapounov optimization [86] or stochastic dynamic programming [85] are also explored. In a more recent survey, primarily focused on the modeling of uncertainties in microgrids, Kumar and Saravanan [109] highlight two-stage scheduling, another category of approach.

Wu et al. [198] present such a two stage scheduling strategy, which is used to mitigate the effect of uncertainties in long horizon (several hours or days) energy management problems. An initial planning, using coarse production forecasts, is obtained by a first optimization stage which may be relatively costly. Then, a simpler sub-problem is used to adjust the decision variables online when short-term, more accurate predictions are available.

Talari et al. [182] study a microgrid optimization with RES and load uncertainties. A stochastic programming method, based on a MILP formulation with two-phases objectives is proposed. The first phase relies on multiple scenario to evaluate the risks due to uncertainties, which is used in the second phase to perform the final decision.

In [97], Kamjoo et al. handle uncertainties in an energy dispatch problem, targeting isolated microgrid, by the way of MOEA. Two objectives are minimized, the first being

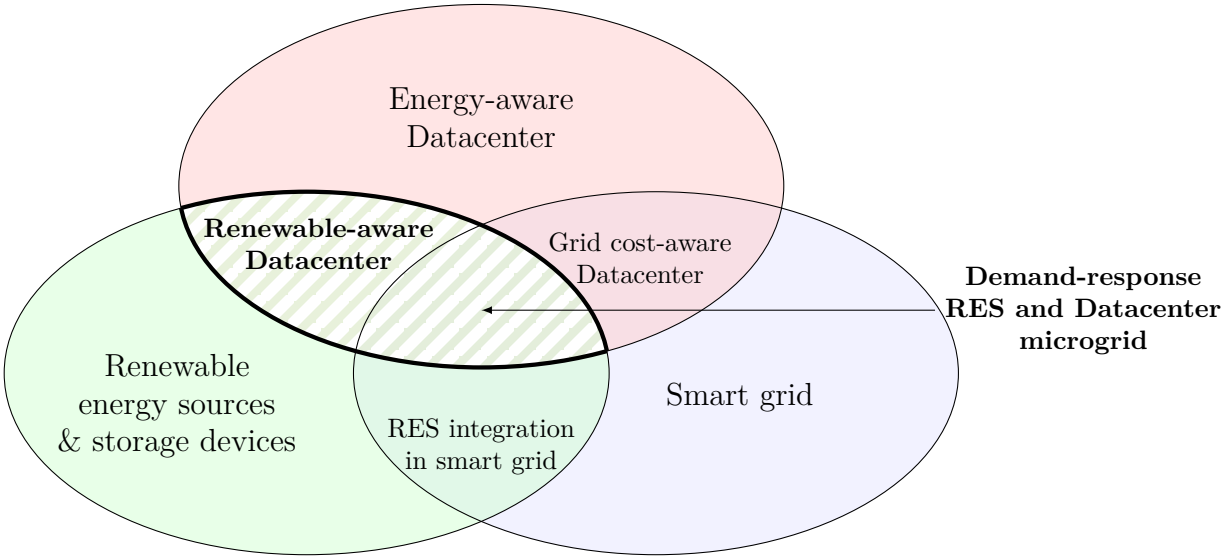


Figure 2.8 – Main related research areas and relationship between them. The white-hatched area highlights the primary focus of this literature review.

the cost of the planning and the other is the probability of not powering the load at some time. Depending on the required reliability and acceptable trade-off with associated cost, any solution from the Pareto front approximation may be chosen.

Another approach based on existing meta-heuristic is proposed by Nikmehr et al. [145], using a particle swarm optimization (PSO) combined with Monte Carlo simulation to account of uncertainties by evaluating multiple scenarios. This is used to optimize the energy commitment in multiple microgrids connected with each others (networked microgrids), each having uncertain renewable sources and loads.

2.4 Management of datacenter powered by renewable sources

As stated previously, reducing the power consumption of a datacenter and improving the use of RES are non-identical problems. The latter is also actively studied from the last decade. While it is a relatively young research area, its foundations come from several other domains, leading to a quantity and diversity of works in this short period. Figure 2.8 shows the relationship between the most important preexisting areas: the studies in energy-aware datacenters mentioned in section 2.2.3, the electrical-engineering researches in RES and ESD and the improvements of the electrical grid which led to the concept of *smart grid* in the early 21st century.

The smart grid defines basically the modernization of the electrical grid to help the integration of many small producers (including RES) and to give a better control on the optimization of the transmission and distribution system. One of the most studied aspect is the so-called *demand response* [175] which aims at controlling the behavior of the power consumers (demand) according to the electrical grid needs, usually through economic incentives. A common use-case is the temporal shift of delayable loads, which helps to absorb the production peaks (when energy is cheap) and to reduce the overall consumption when additional energy would be costly to deliver.

The relationship between demand response in smart grid and renewable-aware datacenter management research areas is dual. Firstly, temporal load shifting is basically a scheduling problem, thus the models studied in some demand response works are close

to models developed in batch task scheduling [131, 204]. Secondly a datacenter being a kind of delayable load, the smart grid community worked on demand response approaches which consider it as the delayable power consumer [130]. Demand response concept is particularly well suited for handling RES intermittent generation, both for distant sources through the electrical grid and for on-site generation.

Historically, this situation led two different researcher communities to study almost independently the management of datacenters powered by renewable energy around 2010. The smart grid community applied the idea of demand response with intermittent RES to datacenters, generally with fine models of electrical components but rough view of machine and workload models. In parallel, the datacenter community extended energy-efficient scheduling to take RES variability into account, with more realistic datacenter models but rough approximations of power systems. This apparently anecdotal fact is highlighted here because it explains some differences in models, methods and vocabulary across the literature. It also shows how young this research area is.

The present literature review includes works from both original communities, as long as renewable energy is explicitly considered (hatched area in fig. 2.8). Despite its youth, datacenter management with renewable energy sources attracted a lot of attention. The survey of Kong and Liu [105] from 2014 presents and classifies dozens of works. It also summarizes the different ways to address the problem and the two main open issues in the community, which are:

1. whether to use *on-site* or *off-site* energy sources
2. whether to exploit *temporal* or *spatial* diversity and flexibility in the datacenter workload.

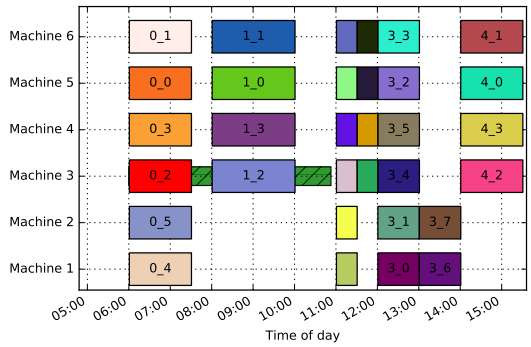
After illustrating the main ideas behind renewable-aware datacenter management, a more detailed classification of the related works is presented and some of the major approaches are described.

2.4.1 Illustration of renewable-aware scheduling

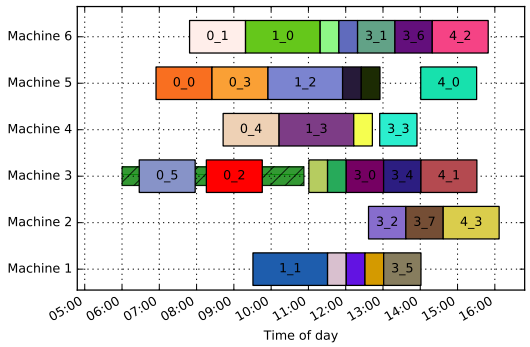
Before digging into the existing approaches, we propose an overall illustration. It aims at highlighting how a renewable-aware scheduler differs from a traditional, energy-efficient scheduler and at providing some intuitions on their behaviors. In this example, we consider a data center powered by a set of photovoltaic panels and additionally connected to the electrical grid. The workload consists in a set of batch tasks with due dates. As the tasks are submitted over time by the users to the data center, they are not known by the system before submission. The scheduler should assign each task to a machine with enough resources to execute it, and has to respect as much as possible the due date required by the user.

With a traditional energy-aware scheduler, the techniques detailed in section 2.2 such as DVFS or power state management may be used to reduce the global energy consumption. For this example, we consider a simple first-fit scheduler, able to power off idle machines. For a given workload, it will execute the submitted tasks as soon as possible, resulting in the execution trace presented in fig. 2.9a. This figure represents, for each machine, the tasks executed on it over time. Each task execution is represented as a rectangle, from its start to its completion. For one of those tasks, named '0_2', we represented additionally the time range between its submission and its due date as a thinner, hatched rectangle. In this example, the task '0_2' appears to end far before its due date, as the first-fit scheduler executes it immediately after submission.

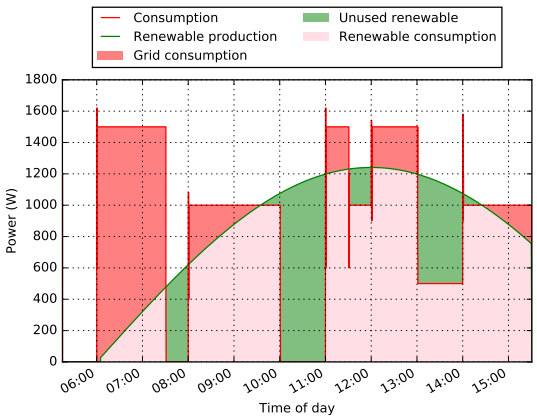
This scheduler is unaware of the origin of the energy used by the data center. Its power consumption for that scheduling, and the production of the solar panels over time,



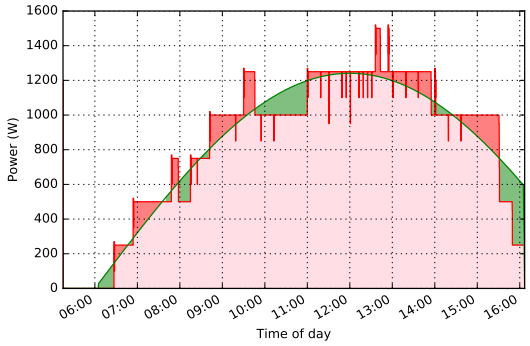
(a) Task execution with traditional scheduler (first-fit).



(b) A possible execution with a scheduling approach aware of renewable energy availability.



(c) Power profile with traditional scheduler (first-fit).



(d) Power profile with a scheduling approach aware of renewable energy availability.

Figure 2.9 – Top: possible schedule of tasks, for each machine, with (b) and without (a) awareness of renewable energy availability. The time between submission and due date is materialized, for the task '0_2', as an hatched rectangle.
Bottom: power profile (production of renewable sources and data center consumption) for a simple scenario, with a scheduler aware of renewable energy availability (d) and with a traditional scheduler (c).

is shown in fig. 2.9c. Obviously, the consumption is not correlated with renewable energy availability. An important part of the energy, in red on this figure, must be drawn from the grid. We also note that a part of the produced renewable energy, represented in green, is not consumed by the data center.

This information can be used by a scheduler that is aware of the renewable energy availability, in order to match more closely both consumption and production. Figure 2.9b gives a possible schedule obtained with the approach presented in chapter 4, for the same workload and also respecting all the due dates. However, almost all tasks are slightly delayed compared to the result of a first-fit algorithm. This is illustrated for the task '0_2', which is executed a couple of hours later, but still ends before its due date. By scheduling tasks this way, the data center consumption over time is quite different from the previous schedule, as we can see in fig. 2.9d. It still does not fit perfectly the renewable energy production, but the consumption and the production are clearly more correlated this time. Note that the total energy consumed by the data center is almost identical in both cases: what changes is how the power consumption is distributed over time and

consequently the source it is produced from.

While it exists as much variations as there are different approaches, most of them are acting in a similar way.

2.4.2 Taxonomy of renewable-aware datacenter management

The literature on renewable-aware datacenter management contains dozens of approaches, but proposing a meaningful classification of these works is a challenging task. One of the main reasons lies directly in the variety of datacenter and electrical infrastructures: many different models are invoked and each model comes with different optimization opportunities and hardly comparable with each other. Another reason is the diversity of the optimization objectives used. At least one of the objective is usually related to the energy, such as the minimization of its total cost or of its ecological impact. At least one other is related to the workload management, maximizing a QoS metric or the revenues. However, the QoS may be measured by many different metrics, as described in section 2.2.1, depending on the kind of workload under use. Consequently, it results a sparse research area, with heterogeneous works in which each author is ultimately working in a *similar but yet different* problem.

Some surveys still attempt to classify the existing works, for instance based on the number of considered datacenters. Kong and Liu [105] separate the approaches using multiple geo-distributed datacenters and the ones with a single datacenter. While the frontier is clear defined, it ignores the fact that many geo-distributed approaches also uses intra-datacenter power management techniques [78, 110]. Another classification is proposed by Shuja et al. [174], based on the involved workload scheduling technique. Three techniques are identified, which are intra-datacenter dynamic load balancing, geo-distributed balancing (*follow the renewable*) and power capping (using power state management and DVFS). It is difficult to assign an exclusive category for every approach from the literature using this classification, as many of them use a combination of several techniques.

More complete taxonomy are detailed in [105, 174], leading to many different dimensions: electricity pricing model, energy storage devices used, carbon emission market model, type and location of RES, kind of workload or objective function. However, none of them gives meaningful exclusive categories which would allow to identify clusters of works addressing very similar problems.

Using the findings of existing surveys [105, 174] and a detailed exploration of the existing literature, a new taxonomy is proposed. Only four main dimensions are derived, each leading to several criteria, shown in fig. 2.10. However, contrary to many conventional taxonomies, most of the criteria (represented as the leaves in the figures) of a single dimension may be combined in a single approach.

Each of the classification dimensions (workload, goal, power leverages and electrical infrastructure) is briefly described below, along with some works illustrating the chosen criteria. A summary of the characteristics of each work is given in table 2.2.

Workload This first dimension foresees the type of workload submitted to the datacenter. The two main classes of workload, batch-like and interactive services already mentioned in section 2.2.1, are used as the main categories. Some approaches are easily associated with one or the other. Gu et al. [78] handle interactive and non-delayable user requests, which is clearly a kind of interactive service workload. On the contrary, Lei et al. [114] follow a typical batch-like workload approach, by considering independent tasks with determined duration and deadline.

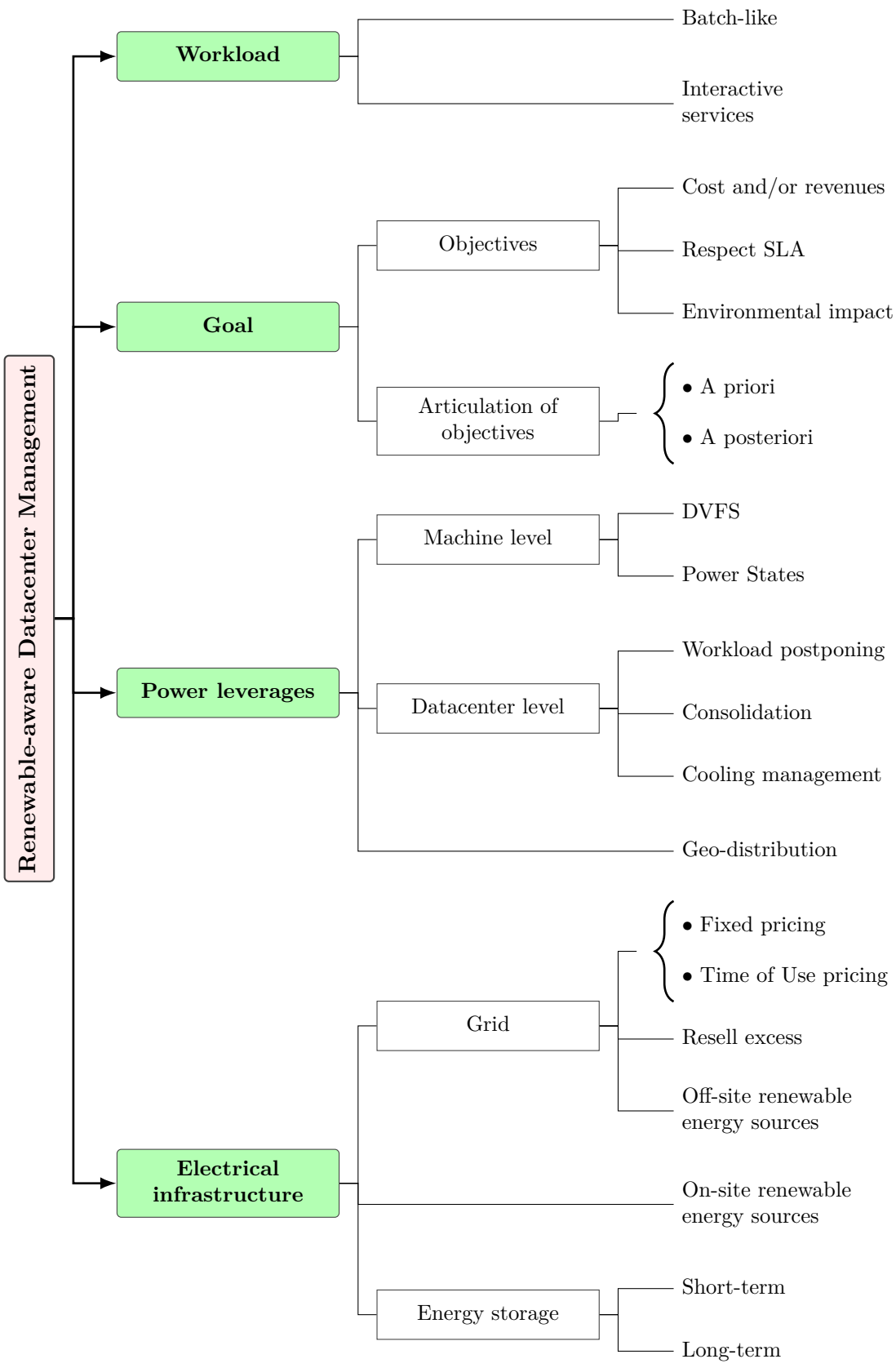


Figure 2.10 – Proposed taxonomy with main criteria identified to characterize related works.

Some approaches consider mixed workload with batch and interactive services such as PIKA from Li et al. [118]. Finally, a handful of approaches are studying a very precise workload, being categorized as batch or services, but showing a possible additional refinement. GreenSwitch (Goiri et al. [71]), which is implemented specifically for Hadoop MapReduce workload but can be loosely considered as batch-like as the processing may be delayed, falls into this category.

Goal This dimension encompasses both the properly speaking *objectives* and their articulation in case there are several explicit objectives. Many specific objectives can be formulated, but they are grouped into three different categories. The first category includes the objectives related to cost minimization or revenues maximization. It is a common kind of objective, as most of the approaches assume a connection to the electrical grid and aims at minimizing the bought energy cost [70]. A few others are taking into account the revenues from the users of the datacenter. Ghamkhari and Mohsenian-Rad [68] deal with profit optimization, calculated as the revenues minus the energy cost.

The second category is the objectives related to the maximization of QoS metrics or minimization of SLA violations. While this seems to be a necessary objective to respect the functional behavior of a datacenter, it is rarely made explicit. Instead it appears as a constraint in the problem formulation [70, 118]. However, the heuristics proposed to solve the problem may not be able to enforce the constraint [110], hence being reasonably considered as an implicit *primary* objective.

Reducing the environmental impact is the main motivation of many renewable-aware datacenter management works. While this may be discussed, we consider here that minimizing the energy bought from the electrical grid, as it is proposed by Li et al. [118] or Kumar et al. [110] for instance, lies in this category. Deng et al. [51] and Gu et al. [78], on the contrary, explicitly propose a minimization of the ecological impact in term of carbon emissions.

A few approaches are considering multiple clearly formulated objectives. Lei et al. [114] propose a multi-objective approach with an articulation a posteriori (Pareto dominance optimization), with four different objectives formulated. Two of them are related to QoS and SLA (minimizing the total makespan, maximizing the satisfaction of SLA) and two others are related to ecological impact (maximizing the utilization of renewable energy, minimizing the total energy consumption).

Power leverages The power leverages are the different ways the management algorithm can act on to reduce the power consumption of a datacenter or delay it. All these techniques can be used alone, or in combination with any others.

The geo-distribution of the workload is the only leverage that exploits the *spatial* flexibility of the workload. Many approaches are investigating the use of multiple datacenters with RES disseminated in different locations, sometimes as the only leverage [51, 78] or in combination with intra-datacenter leverages [110].

Several leverages are available at a datacenter level. Delayable workload, typically batch jobs, can be postponed depending on their deadline or due date, as in PIKA [118]. Virtualization (and containerization) can be used to perform consolidation, by migrating tasks or services to other physical machines and therefore reducing the number of machines required at the cost of higher resource usage on the target machine, as Kumar et al. [110] propose.

Finally, a few techniques can also be used at a single machine or processor level, such as DVFS in the work of Lei et al. [114]. More common is the management of the power

states of the machine (powering off or putting the machine in sleep modes) [70, 71, 118].

Electrical infrastructure This last dimension in the proposed taxonomy concerns the electrical sources and storage devices. Similarly to power leverages, the identified criteria here can be used alone or conjointly with any others in a given approach. The main differences between the electrical models of each approach concern three aspects: the electrical grid (including off-site RES), the on-site RES and possibly the ESD.

The grid is the most common element in existing studies. A single work targeting datacenters, called Blink [171], is considering only RES and ESD with no grid power supply. There are multiple ways to represent the grid nonetheless, each model giving specific optimization opportunities. The possibility to resell a local production excess to the grid provider is explored by some authors using selling price models [78] or considering the grid as an infinite storage device (net metering) [71]. The grid energy pricing scheme varies drastically across the literature. Some works, especially when looking only at minimizing the grid energy consumption without caring for the energy cost are not modeling this aspect or use a static cost [110]. Another common pricing model is the on-peak/off-peak scheme [70, 71, 148], which gives some notion of dynamic energy cost. More advanced pricing models are considered by some authors, with somehow arbitrary evolution of the energy cost depending on the time of use [68, 78].

Fine RES models are occasionally used, such as in the work of Pahlevan et al. [148] with an accurate temperature and irradiance-based PV simulation. In other approaches, the RES production is only modeled as the resulting power, without explicit definition of the underlying models used to estimate these values [44, 119].

The majority of works considering on-site ESD are either using explicitly batteries [71, 118] or some abstract storage devices with model generic enough to encompass other technologies [110]. Using different energy storage devices is rarely explored. Li et al. [115] advocate the benefits of hybrid renewable energy systems for datacenters, including several kind of storage. Some of them (small-scaled batteries and even more supercapacitors) can be viewed as short-term ESD while some others, especially a combination of fuel cells and electrolyzer, are designed to store energy on the long-term, from days to months. However, the approach in [115] is not evaluated with such combination of long-term and short-term ESD. No other approach clearly leverages this aspect, with the closer explored setup being a combination of two different battery technologies in [148].

Uncertainty of renewable energy sources

The handling of energy production uncertainties does not appear in the proposed taxonomy. This is explained by several reasons: the small number of works addressing uncertainties, the variety of methods used making each approach almost unique and finally the holistic aspect of uncertainty management (both goals and infrastructure models are impacted).

Contrary to smart grid management, the field of datacenter management with RES is still at the early stage of considering production uncertainties. Many of existing approaches focus only on the intermittent aspect of RES and do not directly account the uncertainties. Some authors propose ways to mitigate the effects of inaccurate predictions, either by reconsidering their decisions when better predictions are available [70] or by hardening a long term decision with online adjustments taking into account the actual production [148].

Only a few existing works consider these uncertainties in a sufficiently comprehensive way to be considered as a part of the core problem addressed. Deng et al. [50] propose an online control of energy sources (electrical grid and RES) and storage for datacenter,

using two-stage Lyapunov optimization minimizing the operational costs. Although their approach is interesting by some aspects (*e.g.* by requiring no a priori knowledge on the uncertainties), no control is performed on the datacenter itself. It is modeled as a stochastic power consumer, not impacted by any decision, hence making it difficult to classify this work as a datacenter management approach.

Ding et al. [55] study batch job scheduling with uncertain renewable production and electrical prices. The problem is formulated using mixed-integer linear programming (MILP), with multiple scenarios representing a similar sub-problem with fixed value for each stochastic input. These scenarios are created using Monte Carlo sampling of the corresponding distributions and considered equally probable. The optimization objective is a maximization of the expected (sum over all the scenarios) value of a weighted sum of operating cost, carbon emissions and a measure of the risk associated to a given scenario. Because the size of the problem increases with the number of scenarios considered, such a MILP formulation can hardly scale to many scenarios to cover most of the possible realizations of random variables.

2.4.3 Case study of existing works

The proposed taxonomy, along with the examples from the literature, gives an overview of the diversity of the existing approaches. A closer look to some of the works used to illustrate this classification is proposed here to give a brief but more global description of the approach proposed by each.

Gu et al. [78] describe a request scheduler for distributed Cloud datacenters with on-site RES and ESD. It aims to minimize both the energy cost and the GHG emissions. This scheduler is in charge of choosing which node of which datacenter will handle which request, by using one of the several presented heuristics. These approaches are promising for taking advantage of the spatial distribution of datacenters. As this thesis focuses on the management of a single datacenter, we aim to leverage instead the temporal flexibility of the workload.

Li et al. [118] propose PIKA, a scheduling infrastructure able to leverage renewable energy sources in datacenters with a mix of interactive web services and batch jobs. Each job, both batch and interactive, is executed in a virtual machine (VM). While the jobs of the interactive part are executed immediately, the batch ones may be delayed in a queue until enough renewable energy is available or until it must be executed due to its deadline. This approach uses VM migration to minimize the number of physical machines needed to execute the workload at a given time. The unused nodes are powered off to save as much energy as possible.

With their approach named GreenSlot, Goiri et al. [70] considered a small cluster used for scientific computation, and powered partially with solar panels. Using prediction of available renewable power, along with grid electricity price, the GreenSlot algorithm delays jobs to execute them when the cost is the lowest (both in terms of brown energy usage and in terms of purchasing cost). The managed jobs have an estimated execution time and a deadline. This approach uses a greedy heuristic, based on a discretization of future time into *slots* of fixed duration. Each slot is valuated with predicted renewable energy production, grid electricity cost, and number of available computing node at this time. To place a task, the method presented by Goiri et al. consists in testing each possible starting slot. The first one to allow using only renewable energy during each required slot is chosen, if any. Otherwise, the heuristic consists in choosing the one with minimal estimated electricity purchasing cost.

In another paper, Goiri et al. [71] present the GreenSwitch approach, along with a

real datacenter prototype named Parasol in which they implemented and tested their algorithm. This datacenter, power by solar panels and batteries, is a small scale one, containing 64 energy-efficient machines. GreenSwitch is intended to manage MapReduce workloads, and specifically those using the Hadoop implementation. The authors used a Mixed Integer Linear Programming formulation, as a single optimization problem, of both workload scheduling and energy distribution constraints. Several variants of the optimization problem are studied, to explore the impact of using batteries or net-metering. Two different kinds of workload were used: deferrable, which can be assimilated to batch jobs that may be delayed until the end of the experiment window, and non-deferrable, which cannot be delayed. The results presented in [71] show that the GreenSwitch algorithm can significantly increase the use of renewable energy, particularly when the workload is deferrable, compared to a traditional Hadoop cluster.

Pahlevan et al. [148] approach is unique in several aspects. By using an existing energy-aware VM scheduler [102] the authors add a multi-phase energy optimization with RES and energy storage management. The electrical infrastructure contains a PV array, a grid with on/off peak energy pricing and two different batteries banks. These banks are each using a different technology, lead-acid for the first, lithium-ion for the second, in order to leverage each technology in term of price and lifetime. The approach starts by a first offline phase, using irradiance and load forecast to perform a first middle-term (several days) allocation of the energy using relatively costly optimization method. Then, an online phase is repeated every hour to adjust the power consumption and production with the observed values and to perform short-term optimizations, using the initial offline planning as a reference.

2.5 Conclusion and positioning

While the research field on datacenter management with RES is young, many approaches were already proposed to tackle some of the new challenges. As shown by previous sections, numerous IT and electrical infrastructures are studied, each leading to different models and strategies to use renewable energy more efficiently. This diversity, combined with that of possible objectives and ways to articulate them, is highlighted by the proposed taxonomy and the summary of approaches in table 2.2.

Several issues are caused by this diversity of existing works and models. First, this makes it hard to compare fairly the performances of several approaches, as each work is considering a slightly different infrastructure (*e.g.* with/without ESD or dissimilar workloads) and therefore different problems. Then the complexity of the combined IT and electrical infrastructures leads the authors to focus on simplified models compared to the state of the art of each research field (datacenter management and electrical engineering). Finally, this makes more difficult to apply the approaches to real-world datacenters. To be applicable with minimal modifications to a real infrastructure, the model studied in a given approach must be sufficiently close from the target, both in terms of workload type, possible power leverages and electrical sources and storage devices.

For all these reasons, it is interesting to split the overall optimization problem into partially independent sub-problems. The ANR project Datazero [152], for instance, proposes an approach based on independent optimization of electrical infrastructure on one side and IT infrastructure on the other. A central negotiation mechanism is used to manage the diverging objectives of each side and hence ensuring power available matches the requirement while taking into account the quality of service and the ecological impact. With this kind of design, the management of both IT or electrical sides may be viewed as black-

Approach	Objectives	Workload type	Electrical infrastructure	Power leverages	Resolution
Gu et al. [78]	Min. energy cost/carbon emission (QoS constraint)	Services (requests)	On-site RES, grid (ToU pricing, resell), batteries	Geo-distribution	Greedy heuristic
PIKA [118] (Li et al.)	Min. grid energy (QoS constraint)	Batch and services	On-site PV, grid	Consolidation, workload postponing, power states	Greedy heuristic
GreenSlot [70] (Goiri et al.)	Min. non-renewable energy/energy cost (QoS constraint)	Batch (due dates)	On-site PV, grid (on/off peak pricing)	Workload postponing, power states	Greedy heurisitic
GreenSwitch [71] (Goiri et al.)	Min. energy cost (QoS constraint)	Batch (common deadline)	On-site PV, grid (on/off peak pricing), batteries	Workload postponing, power states	MILP
Kumar et al. [110]	Min. grid energy, min. SLA violations	Services (containers)	On-site RES, grid, batteries	Geo-distribution, consolidation, power states	Greedy heuristic
Deng et al. [51]	Min. carbon emission (QoS constraint)	Services (requests)	On-site RES, grid (ToU pricing), batteries	Geo-distribution	Lyapounov optimization
Lei et al. [114]	(A posteriori) Max. renewable usage, min. total energy, min. makespan, max. SLA satisfaction	Batch	On-site PV, grid	Workload postponing, DVFS	MOEA
Ghamkhari and Mohsenian-Rad [68]	Max. net profit	Services (requests)	On-site Wind Turbine, grid (ToU pricing)	Workload postponing (request rate capping)	Interior Point Method (convex optimization)
De Courchelle et al. [44]	Min. grid energy (QoS constraint)	Batch (VMs)	On-site PV, grid, batteries	Workload postponing, consolidation, power states	Greedy heuristics
Blink [171] (Sharma et al.)	Max. QoS (renewable energy constraint)	Services (requests)	On-site RES, batteries	Power states	Greedy heuristics (online control)
Li et al. [115]	Max. battery lifetime, max. renewable energy usage, min. makespan	Batch	On-site Wind Turbine, gas turbine, biomass, batteries, fuel cells	Workload postponing	Greedy heuristics (multi-stage algorithm)
Pahlevan et al. [148]	Min. grid energy (QoS constraint)	Services (VMs)	On-site PV, grid (on/off peak pricing), batteries (lithium-ion and lead-acid)	DVFS, consolidation (based on [102])	Dynamic programming, non-linear programming

Table 2.2 – Summary of main renewable-aware datacenter management works mentioned, using the taxonomy represented in fig. 2.10.

boxes. It allows experts of each field to focus on single-area concerns, hence resulting in more complex and realistic models to be usable. In addition, each resulting optimization module is tied only to one part of the overall model, making easier to apply, for instance, such an IT management approach to several datacenters having very different ESD and RES. Hence it reduces the diversity of the possible models by avoiding the combinatorial effect of joining IT and electrical concerns directly.

This thesis aims, at first, to explore how such a black-box optimization may be performed to tackle the main challenge of RES which is their intermittency. The contribution presented in chapter 4 starts by considering an abstract electrical infrastructure giving limited, aggregated information to a scheduling heuristic. In chapter 5, the concept of black-boxes is applied to both IT and electrical parts.

Another major observation highlighted by this literature survey is the few approaches considering RES uncertainties in a non-trivial way. Hence a third approach is proposed in chapter 6, using detailed information on renewable production forecasts uncertainty to improve the reliability and robustness of the management of a datacenter.

Chapter 3

Simulation tools and data sources

It is a mistake to think you can solve any major problems just with potatoes.

— Douglas Adams, *Life, the Universe and Everything*

Contents

Researches on datacenter energy management in general make heavy use of simulations. They allow researchers to work with large infrastructures which would be extremely costly, to perform experiments faster than real-time and globally improve the reproducibility of the works. Most of the approaches from the renewable-aware datacenter management literature are evaluated only by simulation.

The addition of the electrical infrastructure, on-site renewable energy sources (RES) and energy storage devices (ESD) makes even more difficult to use a real testbed. Very few existing datacenter are currently built with such electrical sources and none are publicly shared for research purpose. Studying weather-dependent renewable energies also adds a level of difficulty for reproducibility: the observed weather condition during an experiment.

This chapter aims first to present existing simulation environments and the specifities required when on-site RES are taken into account. For the purpose of the approach presented in chapter 4, several improvements to an existing datacenter simulator were implemented and are described next. A simulation environment, however, is useless without data. Two very different kinds of data are therefore needed. The first is the production of renewable sources, which is specific to this class of approach. The second is common to all datacenter management approaches, consisting in the workload submitted. Different publicly-accessible datasets are presented and compared for this purpose.

Replaying workload traces directly is subject to many practical issues and only a handful of recent public datasets are available. However synthetic generation of realistic workloads focused few attention along the previous years. In the second part of this chapter, one of the largest public trace, from a Google production cluster, is studied. After analyzing some of its core characteristics, a synthetic generator is proposed to create custom workloads with similar behaviors.

3.1 Datacenter simulation with rich electrical infrastructure

There are basically two ways to evaluate a datacenter management approach without a real testbed: an ad-hoc numerical simulation or using a more generic simulator developed by the community. Ad-hoc simulation may be preferred when the overall model is relatively simple, as the cost of a custom and specific implementation is likely lower than the cost of understanding and adapting an existing complex piece of software. The same goes when the model is rather unusual for the community maintaining a datacenter simulator. These two facts explain the large part of the works in the renewable-aware datacenter management literature which are evaluated with custom simulation environment [119, 148, 150].

While being perfectly legitimate in some situations, such ad-hoc numerical simulation lead to several issues. As many researchers spend time rewriting and testing similar software, the overall effort is likely much more important than required. These custom simulator are rarely released or described enough in the corresponding articles to be reused by others. This may also decrease the ease of reproducibility of some works and in some extents their trustability.

In an attempt to address these issues, several datacenter simulation frameworks, with different purposes and designs, have been developed and are maintained by the community. A few examples are SimGrid [26], CloudSim [23] and DCworms [111]. Unfortunately, none of them provide an easy way to deal with rich electrical infrastructures, as found in the case of a data center powered by on-site RES and ESD.

A full simulation of the electrical infrastructure is not always required when it comes to studying RES-powered data center. In some situations, a simple input of tabular data can be enough, as described in section 2.3.2. For instance, using traces of photovoltaic panels power production as an input allows to study power management strategies for a data center powered only by those PV panels. In similar situations, simulating the RES still gives more flexibility in the scenarios considered, as it is possible to study easily sources with different technologies, sizing and configurations. It also helps to consider different environmental conditions (different kinds of weather, seasons or geographical locations) without requiring external tools.

However, simulating the electrical infrastructure becomes necessary when dynamic components are modeled. These components include notably ESD, like batteries for instance. As the state of charge of a storage device depends on its charge and discharge over time, which in turns depends on the energy used by the data center, it is impossible to represent it with pre-calculated input data. Hence any approach which depends on the state of charge a ESD or which aims to act on its charge and discharge states requires some electrical component simulation.

To the best of our knowledge, no previous work attempts to augment one of the datacenter simulator from the community with rich electrical infrastructure RES and ESD. At the contrary, in addition to the many works using ad-hoc numerical simulation, some researchers developed new simulation platforms adapted to these specific needs, such as RenewSim [44] or ReRack [21]. Unfortunately, they struggle to attract enough attention to them to be considered and used by other authors.

In an effort to address all these issues, the work presented here extended the general-purpose datacenter simulator DCworms with the notion of electrical infrastructure simulation. This integration was performed with several goals in mind.

1. The simulator was designed to be relatively generic, allowing to implement most of the simulation models (task execution time, device power consumption or heat

dissipation for example) as plugins. The same is aimed for the electrical simulation, by implementing electrical devices as plugins.

2. As the first target users are computer scientists, the electrical simulation should allow to implement simple electrical models without advanced knowledge of electrical engineering. In the other hand, it should be *possible* to implement more advanced models.

3. The models of the different components of an electrical infrastructure should be easily re-usable in another infrastructure description.

3.1.1 Design and implementation in DCworms

DCworms, for Data Center workload and resource management simulator, is an event-based simulator written in Java and developed by the Poznań Supercomputing and Networking Center (PSNC). It heavily relies on a plugin architecture for implementing the different simulation models (task performance estimation, power consumption, thermal dissipation...) and scheduling policies, making it easily adaptable to the specific needs. It was originally conceived to study the impact of different scheduling policies on the power consumption of data centers. It is therefore a good candidate for studying power management in data centers powered by RES.

The architecture of DCworms, including the modifications detailed thereafter, is represented in fig. 3.1. Based on a given experiment configuration, and using a given set of plugins, the simulator will output several statistics in different forms (human-readable summary, CSV data and charts). The experiment configuration consists mainly in the workload description (in SWF [1] format with optional per-job description with an XML file), the simulator global configuration, and the resources description. The resources description is provided as an XML file, specifying the number, type and organization of the computing resources.

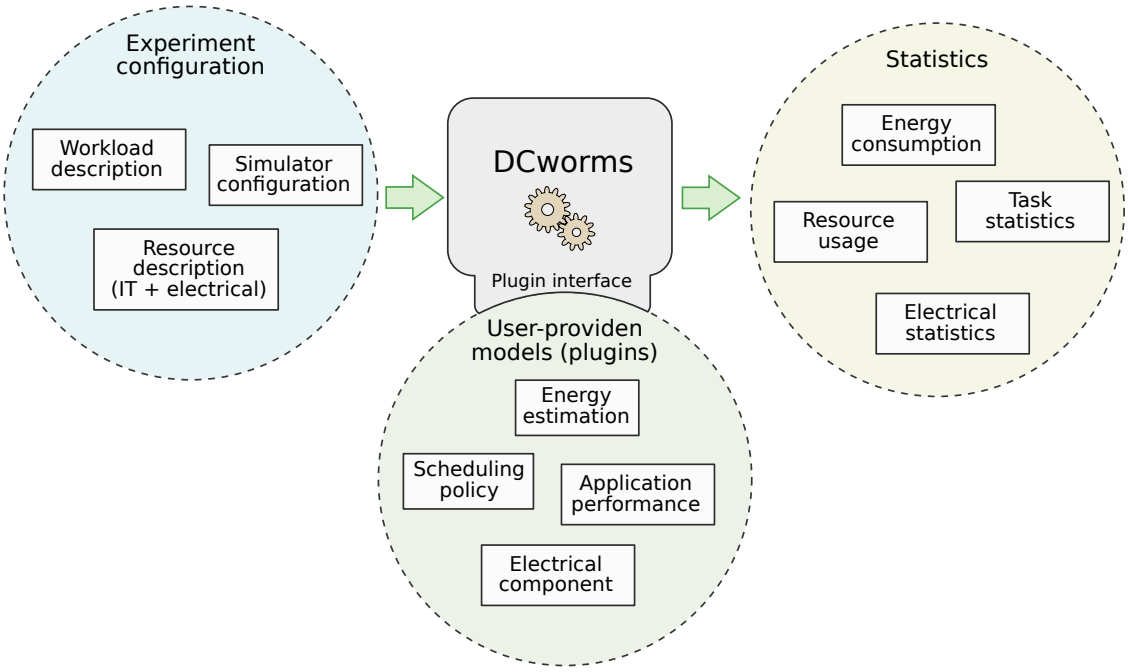


Figure 3.1 – Simplified DCworms architecture with electrical simulation modifications.

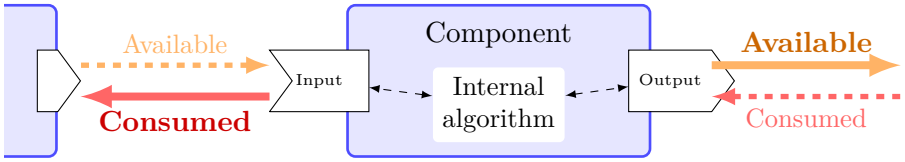


Figure 3.2 – Illustration of an electrical component with one input and one output connections.

Infrastructure model

As stated previously, one of the objectives of this work is to allow to implement simple electrical models with few knowledge in electrical engineering. The goal is not to implement an electrical-circuit emulator, but at the contrary a more abstracted model suitable for an event-based simulation. Hence the proposed abstraction of electrical components is rough from the point of view of electrical engineering works. Accurate electrical-circuit simulation propagates voltage and intensity according to the internal properties of each component (capacitance, resistance, capacitance...) and even simple circuits exhibit complex behaviors at micro or millisecond scale after any change in the circuit (transient states). The model detailed here do not aim to handle such short-time transient states and the components interacts with each other only in term of *power*.

The proposed model considers each individual electrical component as a black-box, controlled by its own internal algorithm. An electrical component can be a physical electrical device (such as a power converter, a battery or a photovoltaic (PV) module), or several physical devices working together and abstracted by a single model.

The component model is represented in fig. 3.2. Each component has one or more electrical *connectors*, which are considered to be monodirectional (either input or output). Those connectors allow to connect any component to another. Each output provides, at any time, the amount of power *available* from it. Conversely, each input indicates the amount of power *consumed* from the output at which it is connected. In a real electrical circuit, the concept of *available* power does not exist: power is simply consumed by a load according to the voltage difference and intensity passing through it. This depends in turn on an equilibrium between its electrical characteristics and the source ones. Using this idea of *available* power permit simpler interactions and simpler component models. The algorithm modeling the component can consider drawing up to the available power from an input connector without risking to be limited at this time. Additionally, the available power can be modeled easily for many common components: its is the maximum production for a RES, the output power after losses at the available input power for an electrical converter or the maximum draw power at a given time for an ESD.

The whole electrical infrastructure consists in the definition of each component plus the description of the connections between them. Figure 3.3 gives an example of electrical infrastructure. In this example, a PV panel and a connection to an electrical provider (grid) are the main sources. They are both connected to a controller, such as a Sunny Island device[1], in charge of managing a battery and of choosing from which source the required power should be draw. Finally, it is connected to the data center supply, which consumes the power required by all the computing resources.

This infrastructure model is very flexible. While it was exploited in the works presented along this thesis, it can handle a large variety of electrical infrastructure, beyond the solely

[1] http://www.sma.de/en/products/battery-inverters/sunny-island-60h-80h.html

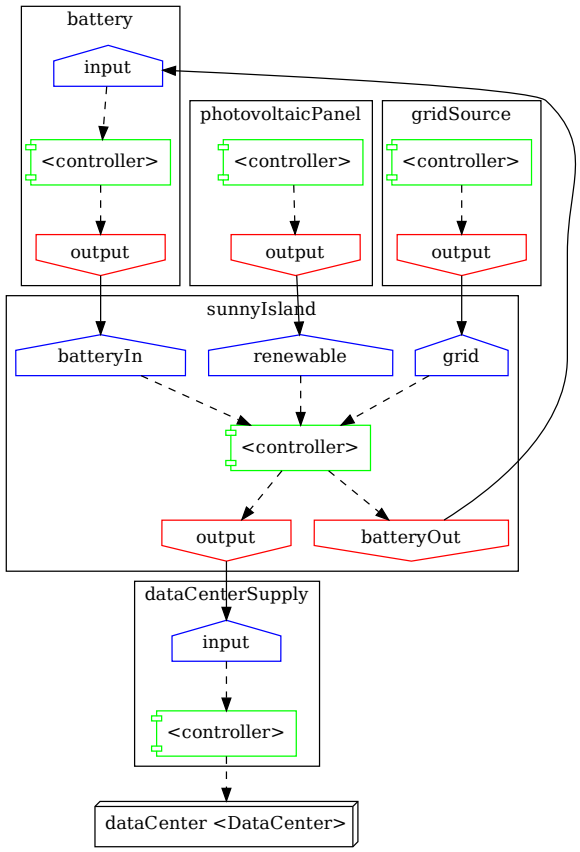


Figure 3.3 – Example of electrical infrastructure simulated with DCworms (including a PV module, electrical grid input and battery). Each black box is an electrical component, the red parts are their outputs and blue ones are their inputs. The arrows indicate the flows of the electrical power between the different parts.

scope of on-site renewable energy sources. For instance, the power supply can be managed at a rack or node-level and may model both Uninterruptible Power Supply (UPS) and redundant electrical networks commonly found in large scale datacenters. Hence, it can be used for studying the robustness to power failures. It is also possible to represent several distinct electrical infrastructures. This can be used for hybrids datacenters, with some machines powered only by renewable energy while having a few racks for critical loads that may be powered by the electrical grid. Another use case is the simultaneous simulation of several geographically distributed datacenters.

Modifications in the simulator

The implementation of this electrical simulation model in DCworms required a few deep modifications in the core of the simulator code to add the main electrical concepts and extend the plugin interface.While the detailed implementation depends highly on the software architecture of DCworms, the main steps are likely to be similar for other existing event-based datacenter simulator. Therefore, they are briefly described here, at the attention of other researchers targeting another software.

At the core of the simulator, the electrical resource management requires to add new types of events to represent changes in electrical state. These events may be generated by any electrical component, such as primary power sources and battery, based on a timer or a change in environmental conditions. It can affects either the available power at an output or the consumed power from an input. They are propagated through the corresponding connector of the origin component (plain arrows in fig. 3.2). The next component should handle this change based on its internal model, in order to satisfy its own constraints between internal state, available power and consumed power. This may be in turn propagated to its own connectors.

In our implementation, each type of component is modeled by its own Java class, leveraging the plugin interface of DCworms. The resource description, initially containing the machines and rack details, was extended to describe the components and the links between their connectors. An illustrative sample is given in listing 1, which instantiates a PV component implementing the model described in section 2.3.2.

In order to be useful for a datacenter management algorithm, electrical component can expose arbitrary interfaces for either providing data or control. For instance, renewable electrical sources expose instant power production and predicted power for an upcoming time window. A battery management device may expose the State of Charge (SoC) of the battery and a way to control its charging or discharging state.

In addition, an automatic monitoring of each connector is performed by the simulator. It allows to retrieve the power flow over time at any point of the electrical network after a simulation, for statistics purpose. Components may also register custom metrics to be monitored in the same way, such as the SoC of an ESD.

```
<electricalResource class="ElectricalCustom" name="photovoltaicPanel">
  <profile>
    <electricalProfile>
      <electricalPlugin>
        <name>experiments.models.electrical.PhotovoltaicPanelPlugin</name>
        <parameter name="srcPower"> <value>435</value> </parameter>
        <parameter name="powerTemperatureCorrectionFactor"> <value>−0.0038</value> </parameter>
        <parameter name="units"> <value>8.0</value> </parameter>
        <parameter name="weatherProvider"> <value>localWeather</value> </parameter>
      </electricalPlugin>
    </electricalProfile>
  </profile>
  <connectors>
    <output>output</output>
  </connectors>
</electricalResource>

<electricalNetwork>
  <connection output="photovoltaicPanel#output" input="converter#renewable"></connection>
  <connection output="converter#output" input="dataCenterSupply#input"></connection>
</electricalNetwork>
```

Listing 1 – Sample of an electrical infrastructure description for the DCworms implementation.

3.1.2 Realistic data for electrical sources

Simulating the production of renewable sources requires some input data. Two main strategies are available for realistic setup: using existing traces from real power plant

or using weather data combined with models such as the one previously presented in section 2.3.2.

Traces of real power plant are sometimes made available by either its operator or public institutions. For instance, the french company Engie provides production data of a wind-turbine plant located near Nancy in France[2]. In addition to the power production for each wind turbine, a few meteorological conditions are measured, all with a 10 minutes resolution. Another example is the experimental Photovolta micro-plant[3], used for teaching purpose at the Nantes University. Despite its small-scale, the various data is collected with small temporal granularity (5 minutes), including the power production and environmental conditions, making it interesting for researchers.

Weather data, such as wind speed vectors, temperature and irradiation, can be obtained online from various public datasets. The most notable is the Modern-Era Retrospective analysis for Research and Applications (MERRA2) [17]. Based on satellite observations and state of the art models, dozen of weather metrics are provided for the whole world, at a spatial resolution of roughly 50 km and one hour frequency. It consists however in low-level weather data, which is difficult to manipulate without good knowledge in climatology. Some other datasets are easier to understand and to use for non-experts, such as the National Solar Data Base (NSRDB) [169], containing photovoltaic-related data for the whole USA. It combines several data sources, including MERRA-2 and ground-stations to obtain more accurate values, reaching 4 km and 30 minutes precision.

There are also a few publicly available synthetic production data obtained using advanced weather and sources models for various locations. For instance, the National Renewable Energy Laboratory (NREL) provides Solar Power Data for Integration Studies[4]. The same laboratory is at the origin of a similar dataset for wind-turbines: the Wind Integration National Dataset (WIND) [57], part of the wind prospector project. Such synthetic traces are an interesting trade-off between real traces and having to implement RES model to output production based on weather data. They provide much more flexibility than the first, by being generated for many different locations different technologies. They also avoid many common pitfalls of model-based production generation by being handled by researchers in weather and renewable energy sources. Therefore, some behaviors such as the low-pass filter caused by large photovoltaic arrays or stochastic effects observed on sub-hour irradiance levels are modeled.

A comparison of the datasets discussed in this section is shown in table 3.1.

3.1.3 Workload data

One of the key element needed to reliably evaluate datacenter management approaches are precise, realistic and adapted workloads. Most study use workload extracted from particular infrastructures [89]. For the purpose of evaluation of scheduling, the mostly used way to take advantages of these data is to directly replay the traces, reducing the possibilities of test to a specific case with particularities or to study an infrastructure with a different scale. This also restricts the set of usable traces depending on the infrastructure and workload model of the approach to evaluate. Another way is to create a synthetic workload, based on characteristics previously observed or expected in datacenter with the targeted purpose.

Several workload datasets are publicly available. For instance the Google cluster data [161], which focused a lot of attention recently because of its large scale: more

[2]Engie Open Data: https://opendata-renewables.engie.com
[3]Photovolta: http://photovolta.univ-nantes.fr/
[4]NREL Solar Power Data for Integration: https://www.nrel.gov/grid/solar-power-data.html

Data	Characteristics: Location	Duration	Granularity	Production: PV	WT	Forecast	Weather: Irradiance	Wind	Temperature	Sky clearness
Photovolta	Nantes	Since 2011	5 min	✓			✓	≈	✓	
Engie Open Data	Vaudeville	Since 2013	10 min		✓			✓	✓	
NSRDB v2 [169]	USA⋆	Since 1998	30 min				✓			✓
MERRA-2 [17]	World⋆	Since 1980	1 hour					✓	✓	✓
NREL WIND [57]	USA⋆	5 years	5 min		✓	✓		✓		
NREL SPDI	USA⋆	1 year	5 min	✓		✓				

Table 3.1 – Summary of the characteristics and type of data contained in publicly available datasets usable for renewable energy sources simulation.
⋆: dataset with many locations in the given region
≈: only partial data is available, insufficient for modeling RES production

than 10,000 machines during a month, with millions of tasks. It was analyzed by many authors [6, 53, 162], and used directly for evaluating some renewable-aware scheduling approaches [101]. In the context of High Performance Computing (HPC) and grid computing, many traces are available through the Grid Workloads Archive [89] or the Parallel Workload Archive[5]. Unfortunately, most of these workloads date of a decade or more, not necessarily reflecting the characteristics of modern environments. For interactive services, other datasets comes from monitoring the requests performed to existing services (usually some website). This includes the 1998 World Cup website traces [10] and the Wikipedia request workload[6] [185].

Synthetic workloads are more flexible than direct trace replay, but require an analysis of these real datasets in order to model their main characteristics. While many works attempt to understand the real workloads [100, 124], they are rarely sufficient to generate similar ones. The contribution presented in the next section consists in an analysis of the Google cluster dataset focused on synthetic generation.

3.2 Analysis and generation of realistic workload

In this section, an analysis of traces from a large scale production environment is proposed. It leads to a model of the corresponding workload, used to create a synthetic generator to create similar traces with any variation in scale or with slightly different characteristics. The traces used as reference are from the Google cluster [161], which contains millions of tasks from an heterogeneous production and development infrastructure. By using the configuration possibilities of the proposed model and implementation it becomes possible to simulate large scale workload.

After describing the Google Cluster workload traces, a review of existing works analyzing the same traces or comparing them to other datasets. Then, a generic model for batch-like workload is detailed and is applied to the traces in order to determine their distributions and characteristics. It is followed by an automatic clustering of the tasks to provide an alternative classification of the different parts of the workload. Finally, a reusable and customizable workload generator is detailed as a simple algorithm imple-

[5]Parallel Workloads Archive: http://cs.huji.ac.il/labs/parallel/workload/
[6]Wikipedia request traces: http://www.wikibench.eu/?page_id=60

Table 3.2 – Distribution of number of events per task.

Number of events	**1**	**2**	**3**	**4**	**5**	**6+**	**Total**
Number of tasks	248	1,093,992	24,258,907	19,378	17,215	34,991	25,424,731

mented in Python.

3.2.1 Workload characteristics

The data used in this study and analyzed below come from one of the datacenter of Google. Google provided a dataset [161] of some of its servers usage in 2011. The monitored datacenter is composed by more than 12,000 servers with heterogeneous characteristics. The different statistics were collected for a period of 29 days. Data have been anonymized to protect the servers configuration (CPU, Memory, Disk) by being normalized between 0 and 1. The data consists in a zipped collection of files taking 43 GB compressed. It includes six main data tables: *job_events*, *machine_attributes*, *machine_events*, *task_constraints*, *task_events* and *task_usage*. A job is composed by one or many tasks. The present work focuses on the *task_events* table. This record contains 500 CSV files, 25 millions tasks. This record encompass 100 millions events which describe each task and their life cycles. During its lifetime, a task has a state: *Unsubmitted, Pending, Running* or *Dead*. An event indicates a transitions between states (9 events):

The initial task event is *Submit* which allows the task to be scheduled. The *Schedule* event means that a task has been scheduled on a particular server. There are ending events such as *Evict* which means that a task has been unscheduled because of a higher priority task; as *Fail* is when a task was unscheduled because of a task failure; *Finish* means that a task finishes normally; *Kill* is when a task is canceled by a user or a management program or when a linked task died. Finally *Lost* event characterizes information missing because of monitoring system issue.

Considering the 500 CSV files which describe task events, the results show that task pass throws several events. Table 3.2 describe the distribution of number of events for each tasks.

A huge majority (95%, 24,258,907 out of 25,424,731) of tasks have only the three classical events: an initial *Submit*, a *Schedule* one and an ending event. Most other tasks (4.5%) are the one which are submitted, scheduled but do not finish during the time frame of the acquired data. The remaining (less than 0.5%) are tasks that have intermediate events of reconfiguration. In the following, tasks taken into account are the one composed of three events.

The way the tasks are ending is also important. Table 3.3, describes the distribution of finishing state for all tasks with three events. Most (73%) completed normally, 26% are killed (in the *Kill* state), and less than 1% finish in another state. As stated in the description of the workload by Google[161], nearly all tasks that do not finish normally are in this *Kill* state. No additional information allow to know the actual reason that caused the tasks to finish abnormally.

The behavior of these two types of tasks are quite different. As an example, the total time of execution of the 73% of tasks with the *Finish* state are accumulating 13×10^9 seconds (with an average makespan of 1694 seconds) whereas the 26% of tasks with the *Kill* finish state are accumulating 20×10^9 seconds (with an average of 7994 seconds).

Most tasks are composed of three events and are finishing in either *Finish* or *Kill*

Table 3.3 – Distribution of finishing event for all tasks with three events

Events type	**Finish**	**Kill**	**Fail**	**Other**
Number of tasks	17,775,284	6,381,906	86,348	15,369
Percentage	73.31 %	26.31 %	0.35 %	0.03 %

Table 3.4 – Number of tasks and frequency for each scheduling class

Scheduling class	**0**	**1**	**2**	**3**
Tasks number	20,317,398	3,327,283	533,330	49,614
Percentage	83.86 %	13.73 %	2.20 %	0.21 %

states. Only these tasks are considered in the following.

All tasks are not treated equally. Production tasks cannot be stopped or postponed, some can be postponed but not stopped, and a few monitoring tasks can be stopped but their is no more interest to restart them latter if they were postponed. Schedulers or middleware need information on the requirements of tasks in order to evaluate the quality of their decision. One important such requirement is the impact of time on their allocation. In most systems, tasks are labeled with their priority or deadline, affecting the policy of the scheduler.

The Google Workload Traces collection contains, for each task, two properties for their *priority* and *scheduling class*. According to the Google's documentation [161], the priority is used by the cluster's scheduler, whereas scheduling class is used locally by a machine to manage resource usage.

The table 3.4 gives the number of tasks for each scheduling class. This class value is between 0, for the least latency-sensitive tasks, to 3 for the most critical tasks.

Priority level is a number between 0 and 11, with 0 as the lower priority. Table 3.5 provides the count of tasks per priority value.

The figure 3.4 shows the average makespan (log scale) for each priority. The tasks with the highest priority levels (9 and 10) have a significantly longer average makespan than the rest. At the opposite, the priority level 5 has the lowest makespan, but because of their very low occurrences (only 104 tasks) it may be caused by some sort of noise.

For this specific dataset, the priorities are logically grouped in 4 categories[7]. Those

[7]Additional traces documentation: https://github.com/google/cluster-data/blob/master/ClusterData2011_2.md

Table 3.5 – Distribution of the tasks across each priority level.

Priority	**0**	**1**	**2**	**3**	**4**	**5**
Tasks	5,701,546	2,357,274	1,078,476	1,027	13,975,078	104
Priority	**6**	**7**	**8**	**9**	**10**	**11**
Tasks	633,445	0	249,932	230,164	579	0

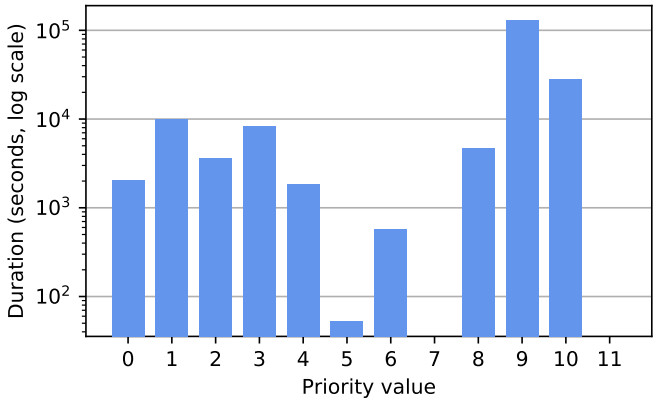


Figure 3.4 – Average tasks makespan for each priority

Table 3.6 – Number of tasks and frequency for the four priority groups

Priority group	**Free**	**Normal**	**Production**	**Monitoring**
Tasks number	8,058,820	15,938,062	230,164	579
Percentage	33.26 %	65.78 %	0.95 %	0.002 %

groups are, sorted by increasing priorities, *free*, *normal*, *production* and *monitoring*. Table 3.6 is obtained by grouping the data from table 3.5 using those priority groups. It appears that, in terms of task count, the *monitoring* group is almost insignificant, with about 0.002 % of the total.

There is no direct and absolute relationship between scheduling class and priority. A task with a high priority may also have a low scheduling class and vice versa. However, the figure 3.5 shows some statistical relationships between those values. For each scheduling class, it gives the distribution of corresponding tasks between the priority groups. Except for the monitoring class, there is a small correlation between priority and scheduling class. The higher the scheduling class, the higher the priority. Concerning the global distribution, most tasks are in the normal scheduling class for each priority.

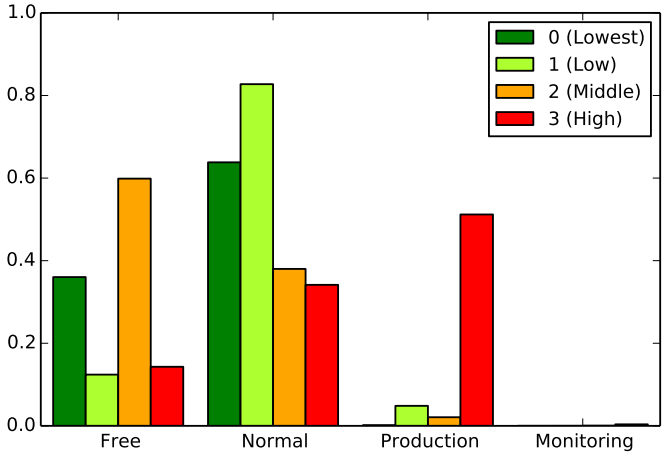


Figure 3.5 – Distribution of priority groups, for each scheduling class

3.2.2 Other studies of the dataset

Several studies have previously focused on a better understanding of different aspects of the Google traces dataset and of the characteristics of the described tasks.

In [163], authors described the heterogeneity of the hardware resources in the Google traces. They classified priorities in five categories, differing compared to the current documentation: Infrastructure (11), Monitoring (10), Normal production (9), Other (2-8) and Gratis(0-1). They identify the *boulders* and *sand* of this workload: they have classified tasks in a majority of small tasks (sand) and an important number of large tasks (boulders). In this analyze, they shown that 2% of tasks represent 80% of CPU, Memory and that 92% are long tasks with a *Free* priority. Reiss et al. [162] also analyzed the Google cluster performance. According to this article, many long jobs have stable resource utilization, which helps the adaptive resource schedulers. They show that machine configuration and workload composition are heterogeneous. These two works give a good understanding of the Google dataset and exhibit the importance of an heterogeneous datacenter to adapt the resources to the demand. However, they are not aiming to help on using this workload in another context.

Gbaguidi et al. [67] evaluates the gap between the requested resources and the one consumed by tasks within the datacenter, highlighting the lack of energy consideration in the dataset. The requested average load for a processor is 10% on the Google datacenter. This Google trace analysis shows that processors are overall under utilized which leads to an increase of the energy consumed: 90% of available processor computing power appear to not be used. Moreover, most of the workload has a low priority and is not sensitive to the latency. It shows that there is only a very few number of sensitive tasks, based on their scheduling classes.

Alam et al. [6] provided a statistical analysis of the traces to make some reference job profiles emerge. They based this approach on resource usage, clustering of workload patterns and classification of jobs with k-means clustering. The trimodal nature of the job distribution is highlighted, both for task duration, and for the resource consumption. Unfortunately, the article does not give any data usable to generate similar distributions, focusing only on the ability to classify the jobs.

In [128], authors analyzed how the servers are managed in the cluster and how the workload behaves during the period of monitoring. They have clustered the machine population per same CPU and memory. Contrary to the finding of [162], it is claimed that the machines are almost homogeneous, with 93% of them having the same CPU capacities and 86% the same memories capacities. A few aspects of the workload behaviors are also explored in this work. The main observation concerns the large amount of jobs that are frequently killed.

Liu et al. [124] analyzed and elaborated the characteristics of the Google traces. They measured some characteristics of the duration of execution (makespan) and waiting time of jobs, but without focusing on the laws driving their distributions. The study is mainly focused on job submission over time, not on the characteristics of the submitted jobs. Based on this analysis they present a workload submission prediction method based on several existing approaches. This contrasts with the work detailed here, which proposes to model the job submission along with their duration and priority.

Di et al. [52] evaluated the Google trace compared to grid computing and HPC systems. They found that the Google dataset have a finer resource allocation with respect to CPU and memory than grid/HPC systems. They compared the CDF (cumulative distribution function) of the job length, 55% of tasks finish within 10 minutes for the Google dataset, which appears to be shorter than other studied traces. In addition, Google jobs

are submitted with higher and more stable frequency than the grid computing jobs. In another work, Di et al. [53] have observed that, for the Google workload, the resource utilization per application (which may contain several jobs) follows a typical Pareto principle. Di et al. used a k-means clustering algorithm based on task events and resource utilization to classify applications. Four classes of applications are identified (*single-task application*, *sequential-task application*, *batch-task application* and *mix-mode application*). Additionally, a correlation is found between some task events and application class. For example, 81.3 % of failed task events belong to batch-task applications. These two publications show that the behavior of tasks in the Google workload, despite the differences observed in submission frequency and task duration, follows many similar laws from other grid/HPC workloads.

All these studies are mainly focused in understanding and characterizing the workload. The work presented in the following does the same in a first step, but with the aim of providing sufficient information for generating similar workloads. It also tries to keep the isolated high-level characteristics, such as the parameters of the distributions used to model the workload, as meaningful as possible. This enables the modifications of these parameters to fit similar workloads differing only by some aspect.

3.2.3 Workload laws

The workload traces are analyzed at an inter-task level, without focusing on the intra-task characteristics such as their resource consumption over time. While the traces contain both batch jobs and interactive services [163], there are no explicit distinction between them in the dataset. Instead, all are represented with a batch-like model, service instances being allocated and stopped dynamically with the same mechanisms than batch tasks. For this reason, characteristics studied here are usually associated to batch workload:

- Starting time: Usually defined as the inter-arrival time between tasks
- Priority: Used to evaluate the relative importance of tasks
- Makespan: Length of the tasks in seconds

To account for the hybrid nature of the workload, a last characteristic is isolated, consisting in the *type* of a task, being either a usual batch-task or a service.

Type

As described in section 3.2.1, most of the tasks end with one of the *Finish* or *Kill* events. In the first case, the task just finishes its work and ends normally. In the second case, the documentation is unclear about which are the possible causes of a *Kill* event.

By analyzing those two sets, it appears that they do not follow the same statistical properties. Figure 3.6 shows the average execution time, depending on both priority and finished or killed groups. Whereas the makespan for the finished group tends to decrease when the priority increases, the opposite is observed for killed group. In addition, the killed group is always longer in average. The difference is the most significant for *production* level, with about 493 seconds for finished group compared to 229288 seconds for killed tasks.

Figure 3.7a shows the effective mass of jobs ending with a *Finish* event. It shows the simultaneous running tasks over time that finish well. The mass follows a fractal structure coherent with the inter-arrival and makespan laws. It also exhibits daily and weekly patterns. For long-term characterization of traces, such long-term patterns should

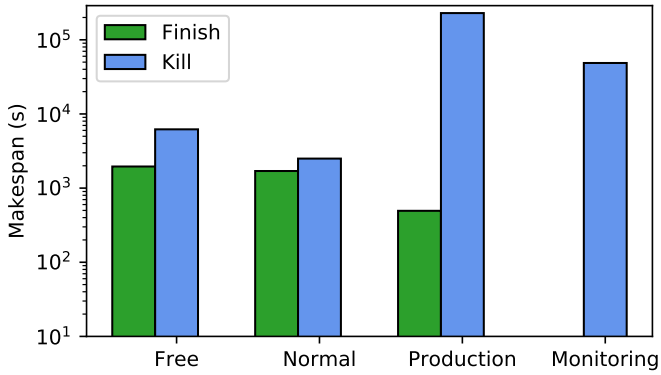
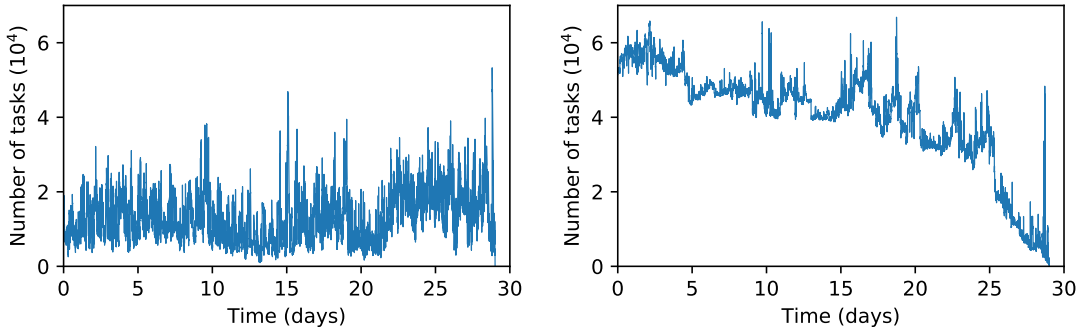


Figure 3.6 – Average makespan for each priority group, depending if the task finished by a *Finish* or a *Kill* event (log scale)



(a) Tasks ending with a *Finish* event

(b) Tasks ending with a *Kill* event

Figure 3.7 – Number of running tasks at each time depending on their ending event.

be taken into account and modeled. Near the end of the monitoring there is a peak, with a high number of jobs. They are executed before the end of the experiment window and due to lack of information it is unknown if it comes from an actual peak of submitted jobs (like near the half of the monitored data) or if it is an artifact due to the end of the monitoring of the cluster.

Figure 3.7b shows the effective mass of jobs ending by a *Kill* event. Contrary to the previous mass, the effective mass of executing jobs with a non-finish event follows a less regular curve. The average makespan of these tasks is higher than of the *Finish* category and thus the mass toward the end of the monitoring window decreases as these long-lasting jobs start finishing out of the monitoring window. Even with a lower number of tasks, this Figure shows that the overall mass is higher due to these longer tasks. The high variability is less linked to daily and weekly timescale, as these jobs are more related to infrastructure services whereas the one ending with a *Finish* event are more often jobs submitted by actual users of the platform.

Overall, the average number of running tasks at a particular time is largely superior to the number of servers. It shows that most tasks use less than 100% of a server resources. Actually, other files in the Google dataset contain information about processor and memory consumption over time for each job and show that these are often not using whole servers.

Those different characteristics, especially the important execution time and the number of high-priority tasks in the killed group, suggest two different populations. Our

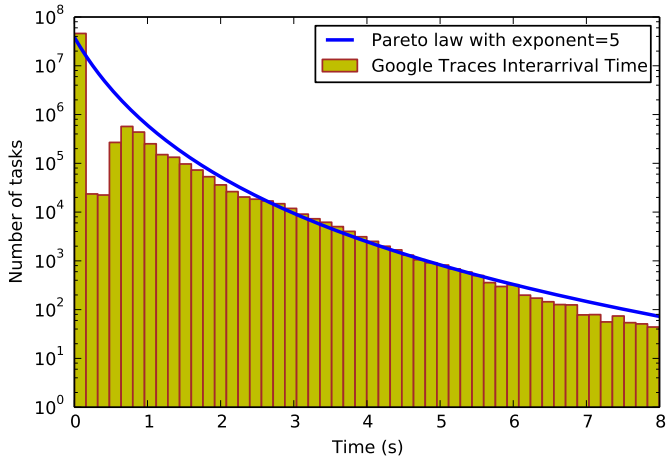


Figure 3.8 – Cumulative distribution of inter-arrival time (s) between task/services (log-scale)

hypothesis is that most of the killed tasks are, in fact, long-running *services*. It may be some monitoring services, web servers, or any other task implementing a service for other internal or external tasks. This kind of service his no pre-determined duration, and may be started and killed to scale with the demand or with the cluster usage. The killed set should also contain some tasks killed manually for other reasons, but it can't explain the amount of *production* tasks stopped this way.

In the following sections, we will use *task* to refer to a finite task, which has a given amount of work to do before to finish normally. As an example, typical kinds of tasks in a such cluster are MapReduce jobs, each containing a set of map and reduce tasks. The word *service* will be used to refer to a task which, at the opposite, has no precise amount of work to do, and therefore just runs until stopped.

Submission Time

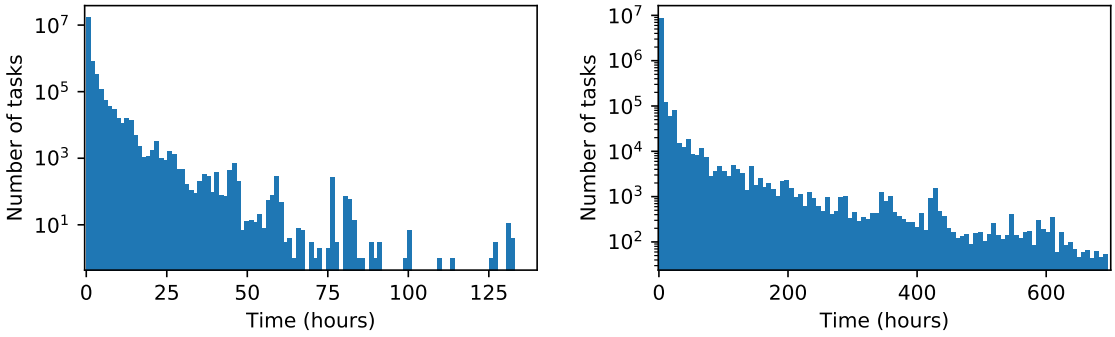
As described in the state of the art [163], tasks and services are submitted continuously on the monitored platform which is highly charged. Very few inter-arrival times are over $10s$, and the average is $0.052s$. The inter-arrival law is well modeled by a Pareto distribution, with a λ parameter set to 5 in the case of the Google dataset as shown on figure 3.8.

A gap for very small intervals can be explained as a large number of jobs, each containing multiple tasks or services, can be submitted simultaneously, breaking the independence assumption. But even with this remark, a Pareto distribution simulates well the inter-arrival time.

Makespan

The execution time of tasks is one of the main characteristics of a workload, as it differs a lot depending on the kind of application it contains. Particularly, the task makespan is clearly different between traditional grid workloads and this cloud trace, as pointed by Di *et al.* [52]. As most other characteristics are similar, a method to have more HPC-like traces would be to increase the average makespan.

Tasks In figure 3.9a, the distribution of finished tasks makespan shows a huge number of small tasks, and few very long ones. The shape is typical of long-tailed distributions,



(a) Tasks finished normally (*Finish* event).

(b) Time before the *Kill* event of killed tasks.

Figure 3.9 – Distribution of the tasks execution time (makespan).

Table 3.7 – Statistics for execution time of tasks finsihed by a Finish event

Priority	Count	Average	Std. dev.	Median
Free	5,081,775	1,937.9	4,518.7	345.7
Normal	12,677,000	1,600.3	5,231.4	492.6
Production	49,190	493.9	1,955.1	39.1
Monitoring	0	N.A.	N.A.	N.A.
All	17,807,965	1,693.6	5,034.5	448.9

like Pareto or log-normal. Table 3.7 gives more precise statistical information. The shape of the distribution is confirmed by the gap between the median value and the average. Standard deviation is also important compared to average, another property of long-tailed distribution.

To use the more appropriate distribution for modeling the makespan, we compared the Kolmogorov–Smirnov (KS) statistic between several fitted distribution and the original data. The more the KS statistic is close to 0, the more the chosen distribution is similar to the real data. A fit with a Pareto distribution gives a Pareto coefficient of 0.113 and a scale factor of 0.073 for these tasks. Its calculated KS statistic is 0.5. The parameters of the fitted log-normal distribution are $\sigma = 1.42$ and $\mu = 6.25$, with a KS statistic value of 0.057. By looking at the KS statistic, the better model to describe this characteristic is the log-normal law.

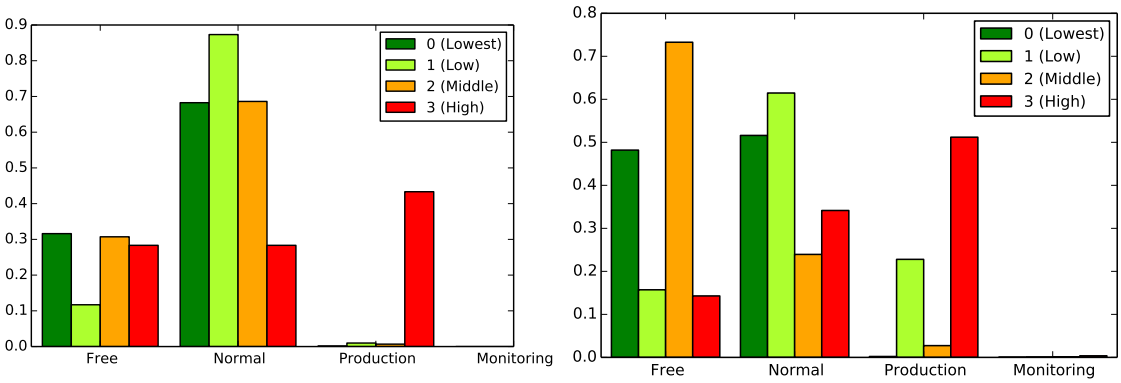
Services The figure 3.9b, similarly, shows the same kind of distribution, with even longer makespans. By comparing table 3.8 with the statistics of finished tasks, it appears that the average makespan is about 5 times higher, with a lower median. A fit with a Pareto distribution gives a Pareto coefficient of 0.0916 and a scale factor of 0.0077 for these tasks. Using a log-normal law, fitting it gives $\sigma = 2.06$ and $\mu = 6.14$. The calculated values of KS statistic are 0.49 for the Pareto distribution, and 0.064 for the log-normal one. For the services too, using a log-normal distribution to model their makespan is accurate.

Priority

The exact semantic of priority and scheduling classes is not specified in the original document. From the combination of priority and scheduling class, importance of tasks and

Table 3.8 – Statistics for execution time of tasks finished by a Kill event

Priority	Count	Average	Std. dev.	Median
Free	2,977,045	5,195.6	24,336.4	595.4
Normal	3,261,062	2,688.5	19,468.9	213.9
Production	180,974	148,424.4	294,182.3	31,429.4
Monitoring	579	388,232.5	579,101.9	71,831.6
All	6,419,660	7,994.3	59,363.8	333.8



(a) For tasks finished by a *Finish* event.

(b) For tasks finished by a *Kill* event.

Figure 3.10 – Distribution of priority groups, for each scheduling class.

services can be deduced. To represent this relative importance in our model, we use a value in the continuous interval $]0, 1]$. The lower this value is for a task, the lower its priority is.

Tasks For tasks, as shown on figure 3.10a, there is a strong relation between priority and scheduling class. To model this behavior, a simple categorization is sufficient. From the statistic of occurrences, 84% are of the lowest priority, 15% of the next one, and the remaining 1% of the highest ones. It can be simulated with an exponential law, truncated between 0 and 1, with a high λ parameter.

Services For services, as shown on figure 3.10b, the relationship between priority and scheduling class is less visible but present. A simple model would be also a truncated exponential law, with a lowest λ value compared to the tasks priority model.

3.2.4 Comparison with other workloads

Such an analysis of this single dataset can only allow us to model the workload for this specific Google's cluster. However, several reasons give us confidence that our model can be used to describe workloads run by other cloud and grid clusters.

Those traces are known to contain multiple and heterogeneous kinds of applications. This is also the case of datacenters used by most Infrastructure as a Service and Platform as a Service cloud providers, and we think this diversity gives good insights on their typical

Table 3.9 – Centroids and analysis of 2-clustering.

Cluster	Makespan	Priority	Sched. class	Frequency	Disparity
0	3508	0.51	0.72	40.5%	17.2
1	2456	4.1	0.11	59.5%	6.2

workloads.

For datacenters running more homogeneous applications, such as HPC jobs, Di et al. [52] have done detailed comparisons between the traces from Google and those provided by several grid infrastructures. They found some differences, such as the lack of periodic patterns in tasks submission over time for the traces we studied here, compared to the grid ones. Particularly, diurnal submission pattern is a common characteristic of datacenter workloads, and may be added to our model to fit better to a typical grid workload.

Di et al. also discovered several similarities which make our model relevant for grid computing. Notably, distributions of makespan have a long-tailed shape in every workload they studied, which is a key characteristic of Pareto and log-normal laws. They observed differences, such as a smaller average makespan, a higher maximum and more short tasks in the Google traces compared to the grid ones. Those characteristics can be obtained, using our model, by adjusting the parameters of the probability distribution functions.

Kavulya et al. [100] analyzed the traces of a production Hadoop cluster owned by Yahoo. They found the same kind of distribution in the Hadoop jobs completion time, and used a log-normal law to model it. Such similarities between several kinds of workloads show that our model, based on a specific dataset, keeps some important characteristics shared by workloads in other datacenter use cases.

3.2.5 Automatic classification

By assuming two main categories, *tasks* and *services*, we worked on *a priori* classification. It allowed us to establish a preliminary analysis on the dataset and to understand some of its characteristics. However, this classification may seem artificial, and it may be interesting to compare it to the result of an automatic classification.

Some previous work already studied those traces with clustering methods [6, 142], but were focused on different properties than the ones covered here. We used a k-means algorithm to classify all the tasks (ending either by a *finish* or *kill* event) by respect to three properties : the makespan, the priority and the scheduling class.

2-clustering

In the case of our initial assumption, we had only 2 classes. Therefore, it is interesting to study the results of an automatic 2-clustering on the same dataset.

The results of the 2-clustering are summarized in table 3.9. In this table, the frequency is the part of the tasks belonging to a cluster, and the *disparity* is the ratio between the average makespan and the median one for a given cluster. The first category seems to be relatively long tasks, with very low priority. The second one is centered on medium execution time, but higher priority. Moreover, this second class represents around 60% of the total amount of tasks, confirming the high quantity of small tasks in these traces.

With $k = 2$, the clustering is far to be perfect. The figure 3.11 shows the silhouette plot using such clustering. With an average silhouette width of 0.55, we can deduce that

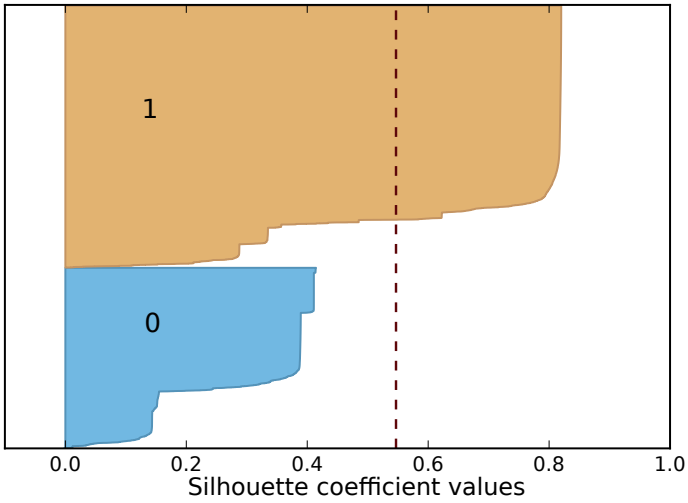


Figure 3.11 – Silhouette analysis of a 2-clustering of tasks using k-means.

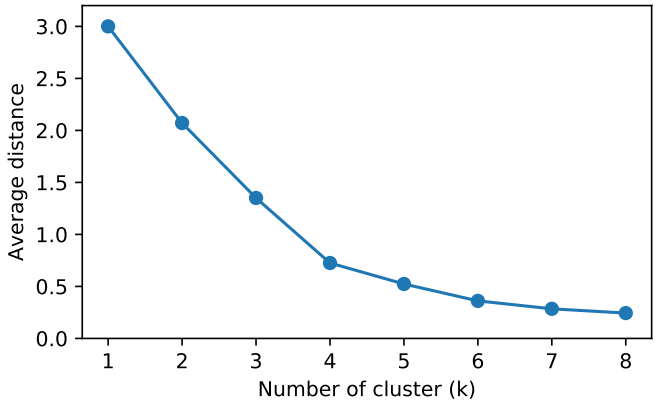


Figure 3.12 – Variation of the average distance to the closest centroid after a k-mean, for different values of k.

those two categories are relevant, without being very accurate. The cluster 0, in particular, gives relatively bad results, probably because it is too wide.

General k-clustering

In order to know which value of k is the most relevant for this dataset, different values have been tested. The figure 3.12 gives the average distance, for each of the normalized data, to the closest centroid for those different k. The elbow method [183] is quite difficult to apply here due to the lack of a clear angle. However, we choose $k = 4$ as a reasonable value: it is still possible to decrease the average distance by using a higher value, but it is a good compromise between number of classes and accuracy.

The results of clustering using 4-means are given in table 3.10. Among the 4 different classes of tasks, the clusters 0 and 1 seems associated to medium tasks with low priority. The first one contains shortest tasks with relatively high scheduling class value, whereas the second one contains longer tasks with very low scheduling class. The cluster labeled 2 contains shorter tasks in average, but with high priority, which still represent more than the half of the total amount of tasks. Finally, the cluster number 3 is quite interesting,

Table 3.10 – Centroids and analysis of 4-clustering.

Cluster	Makespan	Priority	Sched. class	Frequency	Disparity
0	1911	0.9	1.64	17.9%	19.8
1	2482	0.53	0	26.2%	6.2
2	1662	4.2	0.1	55.2%	4.4
3	137630	3.39	0.96	0.7%	1.4

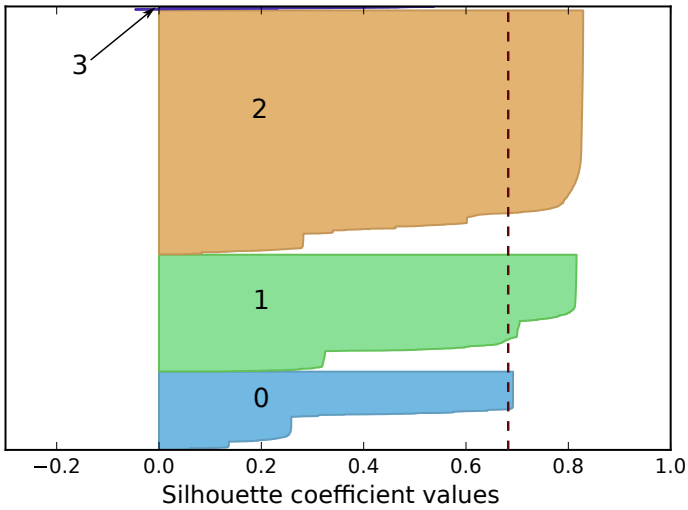


Figure 3.13 – Silhouette analysis of a 4-clustering of tasks using k-means. The cluster labeled 3 contains less than 1% of the tasks and is therefore difficult to see here.

despite the very low proportion of tasks (0.7%) it represents. It consists in the longest tasks (an average of 38 hours), which also have a high priority.

According to the silhouette plot, given in figure 3.13, this clustering seems quite accurate, having an average silhouette width of 0.68. The different classes seem to fit decently the characteristics of the tasks that belong to them, at the exception of the cluster 3, corresponding to the very long tasks.

Our initial, *a priori*, assumption of the existence of two distinct population of tasks and services in those traces may be questioned. However, some of the results obtained by automatic clustering tends to confirm this hypothesis, like the small group of very long tasks in 4-clustering. Moreover, the results confirm that using only 2 categories leads to a decent clustering.

3.2.6 Generating workload

The laws and parameters described in the section 3.2.3 are simulating exactly the same behavior as the workload traces from the Google datacenter. To create more generic workloads it is needed to be able to provide configuration possibilities needed to adapt to different types of experiments and context.

The *Dynamism* is the main characteristics applying to the overall workload. Its value represents the arrival rate of the tasks, without distinction of categories.

Three other characteristics are to be specified for each category (tasks and services) : *Frequency*, *Mass* and *Disparity*

- **Frequency**: Is a value between 0 and 1 representing the percentage of tasks of

```python
import scipy.stats
import random
import numpy

def truncated_expon(lamda):
  while True:
    val = scipy.stats.expon.rvs(scale=1.0/lamda)
    if val <= 1.0:
      return val

def get_makespan(mass, disparity):
  mu = numpy.log(mass / disparity)
  sigma = numpy.sqrt(2*(numpy.log(mass) - mu))
  return scipy.stats.lognorm.rvs(sigma,
               scale = mass / disparity)

def get_next_task(timestampLastEvent, dynamism,
            ratioTask, tasksMass, tasksDisp,
            servMass, servDisp):
  arrival = scipy.stats.pareto.rvs(4, loc=-1)
            * 3.0 * dynamism
  newTimestamp = timestampLastEvent + arrival

  if random.random() < frequencyTasks:
    ttype='TASK'
    priority=truncated_expon(6)
    makespan=get_makespan(tasksMass, tasksDisp)
  else:
    ttype='SERVICE'
    priority=truncated_expon(3)
    makespan=get_makespan(servMass, servDisp)
  return (newTimestamp, ttype, priority, makespan)
```

Listing 2 – Reference implementation of tasks generator implemented in python

this category in the total workload. The sum of the frequencies for the different categories must be equals to 1.

- **Mass**: Represents the average makespan
- **Disparity** : The ratio between the average and the median makespan, must be greater than 1

For these Google traces, the *dynamism*, which is the average inter-arrival time, has a value of 0.05 (in seconds) for Google Traces. As the percentage of tasks in this workload is 70%, we use a value of 0.7 for their *frequency*, and 0.3 for the services.

The *mass* and *disparity* parameters are calculated using the average and median values from the tables 3.7 and 3.8. For tasks, we have a *mass* of 1700 and a *disparity* of 3.8. The values found for services are respectively 8000 and 24.

The implementation of the generator is shown in listing 2 in the Python3 language. In addition to the workloads laws from section 3.2.3, it is also easy to adapt this algorithm and the parameters to generate a workload similar to the k-clustering analysis from section 3.2.5. The values of the parameters can be directly retrieved from tables 3.9 and 3.10.

3.2.7 Additional due date model

Exploiting the temporal flexibility of batch-like workload is one of the keys of renewable-aware datacenter management. As detailed in section 2.2.1, a common model is to have a due date (or a deadline) associated to each task. Unfortunately, the Google traces do not contain this information. It is possible to retrieve the submission date and the scheduling event, but it appears that most tasks are started almost immediately after their submission. This is expected as the Google cluster scheduler does not seem to rely on workload postponing as long as enough computing resources are available.

As we have no knowledge of any work on the modeling of such due dates in cloud workloads, an arbitrary model based on tasks priorities is used. First, we define a task *flexibility* as the time available for scheduling the task with respect to its due date, taking its expected duration into account. The proposed model is parameterized with a value named *flexibility factor*, which is used to modify the magnitude of individual tasks flexibility.

Defining such a metric and studying how it affects the behavior of this kind of scheduling algorithm seems critical for us. It is obvious that the performances of algorithms based on workload postponing depends directly on how much those tasks can be delayed. Because of the lack of existing study, we cannot use known work to take realistic distributions of the flexibility depending, for example, on the kind of data center or the politic of its operator.

Based on this observation, two possibilities exist. The first, used by most of the related study, is to use a workload with arbitrary due dates and analyze the proposed approach on it. The second one, which seems preferable for us, is to propose an arbitrary, parameterized distribution of the tasks flexibility and to confront the approach to different instances of this distribution. The results are therefore easier to extrapolate to real workloads for which tasks due dates are known. In addition, they provide a better overview of the performances of an approach, for a wider range of situations. For those reasons, we strongly believe that defining and studying a flexibility factor results in more interesting comparisons between different approaches. Hence, a model for task flexibility is proposed as a complement of the workload generator, integrating well with it.

Let *flexf* be the flexibility factor used and T_j the expected duration of a given task j (*makespan* in the workload generator). The due date D_j the task j is obtained by eq. (3.1), where f_{min} is a constant used to define a minimum flexibility (possibly 0) and f_{base} is the *base* flexibility. The base flexibility is obtained using a truncated normal distribution with its shape depending on the task priority.

$$D_j = S_j + T_j + f_{base}.flexf + f_{min} \tag{3.1}$$

The task priority, denoted here p_j, obtained through the generator, is a value $\in [0, 1]$. A discrete priority group is determined from this value, using uniform consequent intervals from 0 to 1. By considering G priority groups, the discrete priority g_j of task j is therefore given by:

$$g_j = \min(G, \lfloor p_j G \rfloor + 1)$$

The value of f_{base} is obtained with a normal distribution, truncated at $\mu(g_j) \pm 3\sigma(g_j)$, with different values of μ and σ associated to each discrete priority group.

3.3 Conclusion

In this chapter, an overview of the different aspects considered for performing simulations of datacenter with rich electrical infrastructure was presented. This research area is still

young and present a large diversity of approaches with heterogeneous models, pointed out in section 2.4. Therefore, no simulation infrastructure has emerged yet from the community. Similarly, there is no consensus on common datasets for electrical sources production or workload data which may be used to compare performances of different approaches.

Depending on their needs, their focus and the models used, different choices are made in the evaluation methodology of the works from the literature. It is also illustrated in this thesis. For instance, the scheduling algorithm presented in chapter 4 is evaluated using the improved DCworms datacenter simulator. At the contrary, because of a simpler datacenter model that does not fit well the abstractions of DCworms, some of the experiments conducted in chapter 5 are based on an ad-hoc numerical simulation setup.

A certain diversity of simulation environment in the existing approaches is unavoidable when caused by deep differences in models. Using common tools and datasets should however be encouraged whenever it is possible: it improves the reproducibility and trustability of the results, may help comparing the performances of different works, reduces the engineering cost of developing and evaluating new approaches.

Two contributions presented in this chapter address this point. By augmenting the datacenter simulator DCworms with electrical sources simulation layer, a new tool suitable for renewable-aware approaches is proposed. The workload model and generator based on the Google cluster traces analysis may be used as a base by other researchers looking for a high-level realistic task-based workload.

Chapter 4

Task scheduling with limited knowledge of electrical model

Once men turned their thinking over to machines in the hope that this would set them free. But that only permitted other men with machines to enslave them.

— Frank Herbert, *Dune*

Contents

As detailed along the previous chapters, adapting a datacenter power consumption to fit variable energy availability is one of the keys of an efficient usage of renewable energy sources (RES). Most of the power usage of a datacenter at a given time is tied to the amount, type and distribution of the computing work at that time. Whenever it is possible, being able to change how and when the work is performed is an effective way to adapt the power consumption.

In this chapter, a scheduling approach for batch jobs with due date constraints is proposed and evaluated. It focuses on the *intermittent* aspect of RES production and tackles it by using the *temporal* flexibility of the workload. The goal is to schedule the tasks using preferences of an electrical infrastructure with on-site RES, while respecting the Service-Level Agreement (SLA) from the users. Contrary to existing approaches, described in chapter 2, we propose to separate the optimizations of electrical infrastructure from the optimizations of the computing resources. The scheduling algorithm itself, called Attractiveness-Based Blind Scheduling Heuristic, or ABBSH, is an online greedy algorithm. It exchanges information with an electrical management system, modeled as a black box, in order to take availability of energy in consideration. Along with this algorithm, several multi-objective functions are proposed to handle the trade-off between energy and performance considerations.

The approach has been implemented in the data center simulator DCWorms, using the modifications detailed in section 3.1 to simulate the electrical components. To evaluate the proposed algorithm with realistic use cases, the workload generator presented in section 3.2

is used to mimic the statistical distributions of a real, large-scale cluster owned by Google. In addition, we have evaluated how the choice of the due date of the tasks impacts the performances of our algorithm. Finally, we compared the results of our approach to those of the GreenSlot scheduler [70], a renewable-aware approach from the literature.

The main contributions presented in this chapter are:

- a new scheduling heuristic for data center powered by renewable energy sources, with limited knowledge of the electrical sources infrastructure, detailed in section 4.1.4
- an evaluation of the relationship between the constraints of QoS and the performances of scheduling algorithms for batch jobs, based on the results of our experiments in section 4.4
- a method for computing a lower bound of the energy consumed from the grid to schedule a particular workload, presented in section 4.2.4

In a first part, section 4.1 contains a description of our approach, including an overview of the concepts used and a detailed description of the scheduling algorithm. The methodology used for validating our approach is described in section 4.2. Section 4.3 contains the results of our experiments, which are interpreted and discussed in section 4.4. Finally, the conclusion relative to this approach are presented in section 4.5.

4.1 Proposed approach

The problem addressed in this chapter corresponds roughly to the illustration provided in section 2.4.1: the production of some RES varies across time and batch jobs may be delayed to some extent in order to consume resources (and therefore power) when enough renewable energy is available. Before to detail the models of power sources and computing devices, a brief overview of the global infrastructure is proposed. Then the way electrical and IT objectives are computed and normalized (attractiveness) is detailed. Finally, the scheduling algorithm itself is presented, along with several methods used to handle the multi-objective aspect of the problem.

4.1.1 Data center infrastructure

One of the main goals of the presented approach is to keep separate, as much as possible, the electrical concerns from the computing ones. Therefore, we consider two distinct management systems. The first one is in charge of managing the electrical infrastructure (power sources, storage, and power distribution elements), in order to minimize the power losses and to maximize the usage of renewable energies when a given amount of power is to be supplied. The second one is the computing resources (shortened IT) management, which includes the data center scheduler and the management of the power states of individual machines.

In order to reach a better global optimization of energy use, each management system needs to exchange pieces of information with the other. For instance, with some knowledge about the available renewable power during the near future, the computing management system may adjust the data center consumption in order to bring it closer to what will be available. To keep a low coupling between the two parts, the information exchanged is limited to a small subset. A cost indicator, named *attractiveness*, is used to abstract the optimization objectives of each system to the other, and will be presented in section 4.1.3.

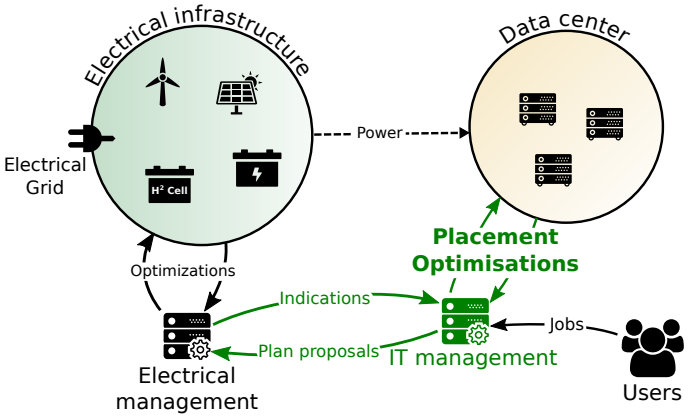


Figure 4.1 – Representation of the proposed infrastructure for both computing and electrical parts of a data center. *(This figure uses icons under CC-BY license, authors: Freepik, Madebyoliver and Yannick Lung.)*

This design, compared to a single optimization loop in charge of both electrical and IT management, allows more flexibility in the considered infrastructures. Indeed, a change in the electrical components themselves or in the electrical optimization scheme would not impact the computing resources management, and vice versa.

An overview of the proposed infrastructure is shown in fig. 4.1. The presented work is mainly focused on the IT optimization part, highlighted in green. Therefore, the electrical infrastructure and its management system are highly simplified in this chapter. However we show the flexibility of our approach regarding the electrical part by evaluating two different objective functions used in the electrical management system: the first considering only the amount of non-renewable energy required and the second focusing on the total energy cost.

4.1.2 IT and electrical models

Across this section, we will detail formally the models used in the presented approach. As our work is focused on the IT part, the electrical model used here is intended to be simple and illustrative.

Data center model

We consider $\mathcal{J}$ the set of the J tasks submitted to the system at any moment. For each task j, we have the following information: its submission date S_j, its due date D_j, its execution time T_j, the number of processors required $n_j \in \mathbb{N}^*$ and the amount of RAM needed r_j. The execution time T_j is normalized by considering the slowest hardware among the machines. In addition, the date at which the task starts its execution is noted B_j.

The computing resources are a set $\mathcal{M}$ of M machines, which may be heterogeneous. Each machine m is associated to its number of processors[1] N_m, the relative computation speed of the installed processors C_m, and its amount of usable memory R_m. The relative computation speed is used to compute the execution time of a task on this processor, such as the time needed for a task j is given by T_j/C_m. The power consumption of a machine

[1]We consider a processor as a physical execution core, assuming no logical core using SMT technologies. In addition, we make the assumption of an homogeneous set of processors for a given machine.

m is obtained using a linear model. Its idling power is noted P_m^{idle} and its maximum (full load) consumption P_m^{max}.

The use of a machine m over time is represented by several functions. The number of processors used at time t is given by $up_m(t)$ and the memory used by $ur_m(t)$. Each machine may be either powered on, off, or may be booting up or shutting down, which is represented by its state $s_m(t) \in \{On, Off, Boot, Shutdown\}$. For managing transitions between on and off states, additional information is needed for each machine, using the power state management model illustrated earlier in fig. 2.6. The booting up time $t_m^{off \to on}$ and the averaged power consumption during boot $P_m^{off \to on}$, with similar information for shutting down, respectively $t_m^{on \to off}$ and $P_m^{on \to off}$. It is then possible to get the instantaneous power consumption of a machine at any time, $p_m(t)$, such as:

$$p_m(t) = \begin{cases} P_m^{idle} + \frac{up_m(t)}{N_m}(P_m^{max} - P_m^{idle}), & \text{if } s_m(t) = On \\ P_m^{off \to on}, & \text{if } s_m(t) = Boot \\ P_m^{on \to off}, & \text{if } s_m(t) = Shutdown \\ 0, & \text{otherwise} \end{cases}$$

We can therefore obtain trivially the total consumption of the machines of the data center with eq. (4.1).

$$P(t) = \sum_m p_m(t) \tag{4.1}$$

A task is executed on a single machine (no migration), and without interruption (no pause and resume nor preemption). Given $\mathcal{E}$ the set of couples of task and machine on which it is executed, we have the following relationships.

$$\begin{aligned} \forall j, \exists m \quad & (j, m) \in \mathcal{E} \\ \forall j \quad & (j, m1) \in \mathcal{E} \wedge (j, m2) \in \mathcal{E} \iff m1 = m2 \end{aligned}$$

To link the task execution model to the use of resources, we define some relationships.

$$\mathcal{R}_m(t) = \left\{ j \mid (j, m) \in \mathcal{E} \wedge t \geq B_j \wedge \ t < B_j + \frac{T_j}{C_m} \right\} \tag{4.2a}$$

$$up_m(t) = \sum_j^{\mathcal{R}_m(t)} n_j \tag{4.2b}$$

$$ur_m(t) = \sum_j^{\mathcal{R}_m(t)} m_j \tag{4.2c}$$

$$\forall t, \forall m \quad up_m(t) \leq N_m \tag{4.2d}$$

$$\forall t, \forall m \quad ur_m(t) \leq M_m \tag{4.2e}$$

$$\forall t, \forall m \quad s_m(t) \neq On \implies up_m(t) = 0 \wedge ur_m(t) = 0 \tag{4.2f}$$

Equation (4.2a) defines, for a given time, the set of tasks executed on a given machine. Equations (4.2b) and (4.2c) represent the use of processors and memory caused by the execution of the tasks. Those resource usages are additionally constrained by the total available ones, as shown in eqs. (4.2d) and (4.2e). Finally, eq. (4.2f) describes that a machine which is powered off should have neither CPU nor memory load.

In this model, the due dates of the tasks are considered to be part of the SLA agreed by the customers. The respect of the promised quality of service for a task consists in

finishing it before its due date, as formalized in eq. (4.3a). As we use it as a metric, the average SLA violation for a given scheduling is given by eq. (4.3b). Its value is between 0 (no violations) and 1 (all tasks violated their due dates).

$$(j,m) \in \mathcal{E}, \quad q_j = \begin{cases} 1 & \text{if } S_j + \frac{T_j}{C_m} \leq D_j \\ 0 & \text{otherwise} \end{cases} \tag{4.3a}$$

$$SLA_{violation} = 1 - \frac{\sum_j q_j}{J} \tag{4.3b}$$

Electrical model

The electrical infrastructure considered here is composed of an electrical grid power supply and a set of renewable power sources without storage. Although we used solar panels in our experiments, no particular assumption on the kind of source is used in our model.

Let $\mathcal{W}$ be the set of the W renewable power sources. The effective available power at a time t for a source w is obtained through $p^w(t)$, and is supposed to be known only for the current instant. For each source, we have a prediction function $\hat{p}_t^w(t')$, which gives the expected available power for a time t' in the future of current time ($t' \geq t$). The prediction is available only for t within a given time window $T^{forecast}$ from the current instant. The prediction function is considered as a black box, and numerous existing works may be used to compute it based on source characteristics and weather forecast, as much for solar panels [32, 90] than for wind turbines [116, 176].

The grid model allows to represent variations of the electricity price over time, such as on-peak/off-peak pricing. Pricing should be known in advance (within the same time window $T^{forecast}$), and is represented by a function $c^{grid}(t)$, which is the price of a kilowatt-hour of grid energy at the time t.

4.1.3 Abstraction of electrical and IT objectives

Each of the management systems, electrical and IT ones, has only a full knowledge of the their own model, described above. Those two parts have therefore different objectives: the electrical part goal is to reduce either the total cost of purchased energy or the amount of grid energy used, and the IT goal is to schedule each submitted task with respect to the SLA.

To exchange bits of information between them, our approach consists in valuing these objectives in an abstracted and normalized way, named *attractiveness*. An attractiveness value is a real $\in [-1, 1]$ which represents, for a given *proposal*, its benefit or cost for one of the optimization loops. A value of -1 is used for the strongest cost, or an impossibility to accept a proposal, whereas a value of 1 corresponds to the most important benefit possible. The values between allow for any intermediate degree of cost or benefit, with 0 representing a neutral proposal.

We propose attractiveness functions for both electrical and computing parts, which associate a proposal to an attractiveness value, using information from its own model. As the attractiveness value abstracts the internal objectives of each part, it is possible to use it to guide an optimization, without the awareness of the elements considered to obtain it. Those values will be used by the scheduling algorithm to reach the best schedule.

Electrical attractiveness

The proposal sent by the IT optimization loop is to be seen as a part of a consumption plan. It contains an average power needed, the time at which the power is required, and

the duration of the request. The electrical attractiveness is therefore calculated based on these proposal data and on the prediction of the model described in section 4.1.2.

In order to demonstrate the flexibility offered by the use of attractiveness to abstract the internal objectives of each optimization loop, we propose two variants of the electrical attractiveness function. Whereas the first one (named variant A) only considers renewable energy use, the second one (variant B) also takes the grid electricity price into account.

The variant A is detailed in algorithm 1. This function has several interesting properties. It is defined as a piecewise function, depending on the value of Δ_{power}. The two sub-functions, for $\Delta_{power} \geq 0$ and $\Delta_{power} < 0$ are identical at the exception of the constants used as *base*, *maxIncrement* and *rate*. In those functions, the value of Δ_{power} determines the impact of the *maxIncrement* factor. The closer Δ_{power} is to 0, the closer the resulting attractiveness is to *base*. At the opposite, when Δ_{power} approaches $\pm\infty$, the attractiveness asymptotically approaches $base + maxIncrement$. Both sub-functions are monotonic in their definition interval. Consequently, the computed attractiveness is bounded in $[\alpha_{pos}, \alpha_{pos} + \beta_{pos}]$ when $\Delta_{power} \geq 0$, and by $[\alpha_{neg}, \alpha_{neg} + \beta_{neg}]$ otherwise. Finally, the speed at which both sub-functions go from *base* to $base + maxIncrement$ is controlled by the value of *rate*.

Algorithm 2 is the attractiveness function for the variant B. It is similar to the variant A, but using a more complex function to determine the value of *base*, using the average price of the grid energy used (C^{grid}). It depends on three other constants, which are the minimum and maximum price of grid energy (respectively θ_{min} and θ_{max}), and λ. The value of λ, called *price factor* thereafter, determines the variation of *base* between $C^{grid} = \theta_{min}$ and $C^{grid} = \theta_{max}$. This modification impacts the value of the attractiveness when $\Delta_{power} < 0$ (when the average renewable energy production is lower than the requirement). Consequently, the more the grid energy price is expensive, the less the value of *base* is.

Some of the constants used in algorithms 1 and 2 are given in table 4.1. The other constants, which depend on the electrical infrastructure, will be given in section 4.2.3.

	α_{pos}	β_{pos}	α_{neg}	β_{neg}
Variant A	0.6	0.4	-0.7	-0.3
Variant B	0.6	0.4	N.A.	-0.3

Table 4.1 – Values used for constants in algorithms 1 and 2. Values of h_{pos}, h_{neg}, λ, θ_{min} and θ_{max} depend on the considered electrical infrastructure.

IT attractiveness

For each proposed task schedule, the calculated IT attractiveness depends on the schedule start time related to the due date of the task ($t_{due} = D_j - T_j$), as shown in algorithm 3. The purpose of this attractiveness function is to favor an early execution, and, more importantly, to penalize a schedule which leads to violate the due date. The difference between t_{due} and the submission date S_j is named *flexibility* of the task ($flexibility = D_j - T_j - S_j$). Indeed, it represents the amount of time during which the start of the task may be scheduled without violating the SLA. A task with low flexibility gives few freedom to the scheduler for choosing an appropriate time for its execution, compared to one with a higher flexibility. Two additional times are defined in addition to t_{due}, before and after it, named t_{urgent} and t_{late}. They are used respectively to reduce the attractiveness shortly before the deadline, and to encourage a schedule shortly after the

Data:
$P_{available}$: predicted average available renewable power during the requested period
$P_{required}$: average required power during the same period
Result: a_{elec} : electrical attractiveness for this request $\in [-1, 1]$
$\Delta_{power} \leftarrow P_{available} - P_{required}$
if $\Delta_{power} \geq 0$ **then**
    $base \leftarrow \alpha_{pos}$;
    $maxIncrement \leftarrow \beta_{pos}$;
    $rate \leftarrow h_{pos}$;
else
    $base \leftarrow \alpha_{neg}$;
    $maxIncrement \leftarrow \beta_{neg}$;
    $rate \leftarrow -h_{neg}$;
end
$a_{elec} \leftarrow base + maxIncrement.\frac{\Delta_{power}}{\Delta_{power} + rate}$;

Algorithm 1: Electrical attractiveness function for a given consumption request (variant A).

Data:
$P_{available}$: predicted average available renewable power during the requested period
C^{grid} : average grid purchasing cost when renewable is not sufficient
$P_{required}$: average required power during the same period
Result: a_{elec} : electrical attractiveness for this request $\in [-1, 1]$
$\Delta_{power} \leftarrow P_{available} - P_{required}$
if $\Delta_{power} \geq 0$ **then**
    $base \leftarrow \alpha_{pos}$;
    $maxIncrement \leftarrow \beta_{pos}$;
    $rate \leftarrow h_{pos}$;
else
    $base \leftarrow -1.0 - \beta_{neg} - \lambda.\left(\frac{C^{grid} - \theta_{min}}{\theta_{max} - \theta_{min}} - 1\right)$;
    $maxIncrement \leftarrow \beta_{neg}$;
    $rate \leftarrow -h_{neg}$;
end
$a_{elec} \leftarrow base + maxIncrement.\frac{\Delta_{power}}{\Delta_{power} + rate}$;

Algorithm 2: Electrical attractiveness function for a given consumption request, with grid price taken into account (variant B).

due date over one occurring long time after. Table 4.2 presents the values used for the several constants required by algorithm 3 to compute the IT attractiveness.

Data:
B_j : beginning of the proposed schedule
S_j : submission date
t_{due} : $D_j - T_j$
Result: a_{it} : IT attractiveness $\in [-1, 1]$
$t_{urgent} \leftarrow t_{due} - \gamma.(t_{due} - S_j)$;
$t_{late} \leftarrow t_{due} + \gamma.(t_{due} - S_j)$;
if $B_j \leq t_{urgent}$ **then**
  $a_{it} \leftarrow v_{base} + \alpha_{base}.\frac{t_{urgent} - B_j}{t_{urgent} - S_j}$;
else if $B_j \leq t_{due}$ **then**
  $a_{it} \leftarrow v_{urgent}$;
else if $B_j \leq t_{late}$ **then**
  $a_{it} \leftarrow v_{late}$;
else
  $a_{it} \leftarrow v_{bad}$;
end

Algorithm 3: Calculation of the IT attractiveness for a single possible task schedule.

α_{base}	v_{base}	v_{urgent}	v_{late}	v_{bad}	γ
0.2	0.7	0.2	-0.9	-1	0.1

Table 4.2 – Values used for constants in algorithm 3 to compute IT attractiveness.

4.1.4 Scheduling algorithm

The purpose of a scheduling algorithm for batch tasks is to find, for each task j, a machine in $\mathcal{M}$ on which to run the task and a starting time B_j. We will name thereafter a *placement* such a couple of a machine and starting time. Our algorithm, ABBSH (for Attractiveness-Based Blind Scheduling Heuristic), is online, scheduling tasks as and when they are submitted, without knowledge of the future submissions. It is implemented as a greedy heuristic, choosing a definitive placement for a task at the time it is submitted.

To decide this placement, the algorithm compares multiple possible placements. For each one, the IT attractiveness is computed, and the power consumption of the data center is evaluated using eq. (4.1) by adding the resources use generated by this placement. The average power consumption is then used to make a request to the electrical loop, which gives the corresponding attractiveness as described in section 4.1.3.

In order to chose among the possible placements, we are facing a multi-objective problem. Indeed, we would like to maximize both electrical attractiveness (which is an image of the *quality* of the power used) and IT attractiveness (representing the respect of SLA criteria). We decided to evaluate ABBSH using different of the multi-objective methods presented in section 2.1, which are recalled and detailed after.

Algorithm 4 presents the method used to place a task j at its submission, using a given multi-objective function. The value of the time step used is function of the execution time of the task and of a coefficient, such as $\alpha_{step}.T_j$. The time step is also bounded with a

Function *placeTask(j, multiObjectiveFun)*
  placements ← empty list ;
  timeStep ← $\min(\alpha_{step}.T_j, step_{max})$;
  windowEnd ← $D_j + \min(\alpha_{window}.(D_j - S_j), window_{max})$;
  for $t_{cur} \leftarrow timeStep$; $t_{cur} < windowEnd$; $t_{cur} \leftarrow t_{cur} + timeStep$ **do**
    foreach *machine* $\in \mathcal{M}$ **do**
      $t_s \leftarrow$ first time $\geq t_{cur}$ with at least (c_j, r_j) available on machine for a duration $\geq T_j$;
      $P_{required} \leftarrow$ average $P(t)$ between $[t_s, t_s + \frac{T_j}{C_m}]$, including current placement ;
      $a_{elec} \leftarrow$ electrical attractiveness for proposal $(P_{required}, t_s, T_j)$;
      $a_{it} \leftarrow$ IT attractiveness for proposal (j, t_s);
      place ← (machine, t_s, a_{it}, a_{elec}) ;
      placements.append(place) ;
    end
  end
  (bestMachine, bestTime) ← multiObjectiveFun(placements) ;
  schedule j on bestMachine with B_j = bestTime ;

Algorithm 4: Simplified pseudo-code of the single task placement function of the ABBSH algorithm. The parameter *multiObjectiveFun* is a function which takes a list of placements with their attractiveness and returns the couple (time, machine) of the best solution.

Function *onTasksSubmitted(tasks)*
  foreach $j \in tasks$ **do**
    placeTask(j, multiObjectiveFun) ;
  end

`// called` $t_m^{off \to on}$ `before` B_j
Function *onTaskReady(j, m)*
  if $p_m(t_{now}) = Off$ **then**
    startMachine(m) ;
  end

Function *onTaskStart(j, m)*
  executeTask(m, j) ;

Function *onTaskEnd(j, m)*
  $t_{end} \leftarrow t_{now} + \alpha_{reboot}(t_m^{off \to on} + t_m^{on \to off})$;
  if $\forall_t \in [t_{now}, t_{end}], \mathcal{R}_m(t) = \emptyset$ **then**
    stopMachine(m) ;
  end

Algorithm 5: Simplified event-based scheduling and power management algorithm for ABBSH.

maximum, to have its value in $[0, step_{max}]$. Similarly, the length of the time window during which placements are considered is based on time between the submission and the due date, with another coefficient: $alpha_{window}.(D_j - S_j)$. To keep this duration into a known interval, a maximum value is also used, keeping the time window in $[0, window_{max}]$. The values of those parameters as used in our experiments will be given in section 4.2.3.

To reduce the global power consumption of the data center, our approach turns off the machines when they are unused. By having the usage planning of all computing resources in the near future, it is easy to anticipate the necessary boot-up. The overall scheduling and power management algorithm is presented in algorithm 5, in an event-driven form. Then functions *onTasksSubmitted* and *onTaskStart* implement the scheduler itself. The power state of the machines is managed by *onTaskEnd*, which allows to turn off idle machines, and *onTaskReady*. The latter is called before the scheduled execution of a task, to let the time to turn on the corresponding machine if needed. Turning on and off the machines too often may result in some drawbacks, including an increase of the total consumption, because of the additional energy required during those steps. In order to avoid unnecessary reboot, Villebonnet et al. [192] define a "minimum switching interval", which is the duration under which it is preferable to keep the machine idle than to do a complete reboot cycle. In our scheduling algorithm, the parameter α_{reboot} is used for a similar purpose, but is relative to the time required for a reboot cycle. A machine is switched off only if it is expected to be idle for more than $\alpha_{reboot}.(t_m^{off \to on} + t_m^{on \to off})$, with $\alpha_{reboot} > 1$.

4.1.5 Multi-objective methods

The scheduling algorithm needs to choose one of the possible placement for each task. Having two objectives, the IT and the electrical attractiveness values, the considerations of multi-objective optimization discussed in section 2.1 apply here. Four multi-objective methods are explored. The first is a weighted sum of both objectives, commonly used in the literature. The second consists in a weighted sum of the hyperbolic sinus of both attractiveness. Finally, two variants of the *fuzzy-based* approach proposed by Sun et al. [180] are used, each using a different primary objective (either the IT or the electrical attractiveness).

The use of a weighted sum is particularly interesting here, as the attractiveness values are already normalized. This avoids the issues of taking the scale of each objective function into account to decide the weights. The choice of the hyperbolic sinus variant is motivated by the shape of this function. In addition to its symmetry, due to the fact it is an odd function, it also has interesting properties which give more weight to the extreme attractiveness values (close to -1 or 1). These functions are defined in eq. (4.4a) for the simple weighted sum and eq. (4.4b) for the sum of hyperbolic sinus. The weighting is controlled by a parameter $\alpha \in [0, 1]$. Parameter β in eq. (4.4b) is used to control the shape of the hyperbolic sinus transformation. A value close to 0 makes the transformation almost linear in $[-1, 1]$. The greater β is, the more the weight of the extreme attractiveness values are important compared to medium ones.

$$f_{wsum}(a_{it}, a_{elec}) = \alpha.a_{it} + (1 - \alpha).a_{elec} \tag{4.4a}$$

$$f_{wsinh}(a_{it}, a_{elec}) = \alpha.\sinh(\beta.a_{it}) + (1 - \alpha).\sinh(\beta.a_{elec}) \tag{4.4b}$$

The *fuzzy-based* method is basically a ε-constraint enhanced by some a posteriori knowledge of the Pareto front. Figure 4.2 illustrates this method, with a scenario where

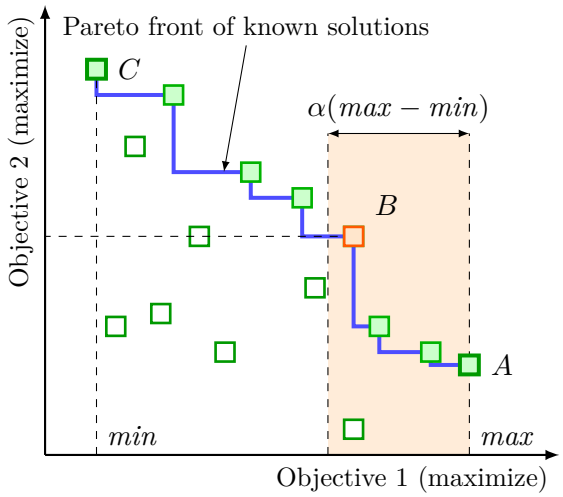


Figure 4.2 – Illustration of the *fuzzy-based* multi-objective approach, with two objectives to maximize. The solutions A and C are respectively the best and worst solution for the first objective. After applying the fuzzy factor, solution B is selected.

two objectives are to be minimized. Initially, the best solution for the first objective is selected (point A in the figure). A constraint on the value of the first objective is then added based on this best value, increased by a certain margin. The final solution is the one minimizing the second objective in this subset (B in fig. 4.2). Contrary to a classical ε method, the constraint relative to the minimum and maximum values observed for this objective. With min_{obj1} and max_{obj1} the extreme values of the solutions on the first objective, the added constraint is:

$$f_{obj1} \geq max_{obj1} - \alpha(max_{obj1} - min_{obj1}) \tag{4.5}$$

The parameter denoted α for uniformity with the other methods, is also called fuzzy factor thereafter.

4.2 Evaluation methodology

Our approach has been developed and evaluated using DCworms [111] as a simulation platform, with the changes described in section 3.1 to handle electrical infrastructure simulation. As detailed along chapter 2, the works studying the management of datacenter powered by RES exhibit a large diversity of models and objectives, making experimental comparisons between them challenging. The proposed approach is compared to the GreenSlot scheduling algorithm [70], which considers a similar IT and electrical model: a single datacenter with on-site RES, grid and a workload composed of batch tasks with due dates. In addition, it has similar objectives (minimization of the non-renewable energy used and of the cost of this energy), allowing a fair comparison of the results.

After describing the simulation environment, the original GreenSlot scheduler and the modifications added either to adapt it to some model specificity or to improve it for the considered workload are presented. Finally, we will detail the parameters used in our experiments for instantiating the models presented in section 4.1.2.

4.2.1 Environment and workload

For the purpose of the presented experiments, we used a simple infrastructure, composed of solar panels and of an electrical grid supply which handle price variation during the day. To keep the experimental setup simple, the solar irradiance data used simulates clear sky days at the equinox and the production is computed using linear-irradiance photovoltaic model given in section 2.3.2.

On the IT side, synthetic workloads are generated using the analysis of the Google workload detailed in section 3.2. As this approach is only focused on batch tasks, we set the parameter $ratioTask = 1.0$ in order to not generate services. The parameters related to execution time are the one extracted from the Google traces: $mass = 1700$ (average task execution time in seconds) and $disparity = 3.8$ (ratio between average and median execution time). The value of *dynamism*, which is the average inter-arrival time, has been adapted to fit the size of our simulated computing infrastructure, and we used a value of 72 seconds.

We also uses the due date model (presented in section 3.2.7), which attributes flexibility to the tasks according the flexibility factor (*flexf*) and the priority generated for each tasks. Three priority classes are defined ($G = 3$): *low*, *normal* or *high*. The parameters of the random distribution (normal law, truncated at $\mu \pm 3\sigma$) for each priority class are given in table 4.3. For the priority class *high*, tasks have only the minimum flexibility, resulting in μ and σ both equal to 0. For our needs, the minimum flexibility value is set to $f_{min} = 60$ seconds, giving sufficient time for the scheduler to start the task.

Using this due date model allows to study in details the impact of the workload flexibility on the results. Specifically, the impact of the flexibility factor on the results of the different heuristics will be pointed out in section 4.3.

Priority class	μ	σ
Low	3600	600
Normal	1200	300
High	0	0

Table 4.3 – Parameters of truncated normal distribution used to define task flexibility (in seconds) depending on its priority class.

In our experiments, each workload starts at midnight, and contains tasks submitted during 72 hours (corresponding to an average of 3600 tasks). As the log-normal distribution used for execution time is long-tailed, some very high values may occasionally be assigned to T_j. The time period considered for the workload being relatively short, we decided to only accept tasks with $T_j \leq 24$ hours (if a task is longer, a new task is generated instead).

4.2.2 GreenSlot

We implemented the "GreenVarPrice" version of the GreenSlot scheduler, which takes into account the variation of grid energy price, based on its description given in [70]. This algorithm is based on a discrete time unit (15 minutes in the original paper experiments), named time *slot*. It considers a finite time window, divided into multiple slots. For each slot, the algorithm keeps the prediction of renewable energy production at that time, the available computing resources, the price of grid energy, and the planned consumption during this slot.

The tasks submitted by the users are put in a waiting queue in Least Slack Time First (LSTF) order. At the beginning of each time slot, the waiting tasks are scheduled in LSTF order. For each task, the algorithm computes the cost for starting the task at the beginning of every slot in the window and schedule the task in order to minimize its cost.

Several adaptations were made in the algorithm presented in the aforementioned paper to fit some differences of computing and electrical models.

- As we have a perfect renewable prediction, we removed the adjustment and re-scheduling in case of prediction error.
- The task execution time is perfectly known, and we do not add tolerance to the time specified at submission.
- Instead of rejecting a job at submission if we cannot respect its due date, we accept it and schedule it after its due date.
- In our implementation, a machine may execute several tasks at the same time (depending on the number of processors installed and required), whereas the original study considered tasks using whole machines.
- In the original article, the idle nodes were put in sleep mode (S3 state). Instead, our implementation turns them off, based on the model presented section 4.1.2.

The core of the GreenSlot algorithm is the slot cost calculation. For a given slot and task, the cost is infinite if the required computing resources are not available during all the task execution time, or if the task is expected to end outside the current window. Otherwise, it depends on the total cost of the grid energy required, plus an eventual penalty if starting the task at this time violates its due date.

Equation (4.6a) gives the cost function, for a given couple of starting slot t_s and task j, when it is not infinite. The set of time slots during which the task is executed is given by S. For a slot s, we have $gridEnergy_s$ the amount of energy which will be required from the grid at this time (in kWh) and $gridPrice_s$ the price of grid energy for this slot (in \$/kWh). A coefficient c_{grid} is used to obtain the cost considered by the algorithm. Finally, an additional penalty may be added to the total cost, as given by eq. (4.6b). This value can be either 0 or $penalty_{violated}$, depending if the due date is respected or not. The values used as c_{grid} and $penalty_{violated}$ are not given in the original work, but we assume the due date violation to be important compared to some grid energy use. We used $penalty_{violated} = 5$, and $c_{grid} = 1$ in our experiments.

$$penalty + \sum_{s \in S} c_{price}.gridPrice_s.gridEnergy_s \tag{4.6a}$$

$$penalty = \begin{cases} 0 & \text{if } t_s \leq D_j - T_j \\ penalty_{violated} & \text{otherwise} \end{cases} \tag{4.6b}$$

When the costs, for a given task, are calculated for every possible starting slot, it is scheduled to begin at the slot with the lowest cost (choosing the earliest if several have the same cost). Once scheduled, the available computing resources and planned consumption are updated for the slots during which the task will be executed.

However, during our early experiments with the GreenSlot algorithm, we found it to be poorly suited to the kind of workload we used. The one used in the original work [70] was a scientific computation workload, with few short tasks and due dates relatively far

away from the submission. Our workloads, at the contrary, contain a lot of short tasks, and some of them have to be started in the next minute.

We identified two causes of bad performances when using our workloads with the original algorithm. Appropriate modifications are proposed to improve it in order to adapt it to our use case.

1. A task with a low flexibility, submitted in the middle of a slot, may need to be scheduled before the next slot to have its due date respected. As the task queue is processed only at the beginning of each slot, such a task is likely to have its due date violated.

2. As the original algorithm considers a time slot to be indivisible, a task which uses only a small part of a slot (either a very short task or the last slot in which a longer task is running) will appear to reserve an entire slot. Because of that, the computing resources usage become fragmented, causing the algorithm to need more resources than really required, or even to delay tasks after their due dates because the nearest slots are already reserved.

To address the first issue, we propose to modify the task submission processing, to schedule it immediately if its due date will be violated before the next slot, instead to put it in the waiting queue. For the second issue, we improve the algorithm to make it able to reserve resources for a fraction of a slot. It is then possible to *backfill* a slot with other tasks, starting immediately after the previous one.

Because of those modifications, we used three different versions of the GreenSlot algorithm in our experiments. The first, referred as *original*, contains none of those modifications, and uses a slot time of 15 minutes (the same value used in the experiments of the authors of the GreenSlot approach). The second, named *partially modified*, uses a shorter time slot (5 minutes) and implements the urgent scheduling modification. Finally, the *modified* version implements both modifications, and a time slot of 15 minutes, as the second modification reduces the need of short time slot. The performances of those different versions will be detailed in section 4.3.2, allowing to argue about the two issues and the impact of our modifications in the GreenSlot algorithm.

4.2.3 Experimental configuration

Before to present the experiment results, we will detail the model instance we used and the values of the various parameters mentioned previously.

IT model We simulated a small data center, quite similar in peak consumption and computing power to the one used by the authors of [70]. To use realistic values of power consumption, we chose an energy-efficient commercial rack server as a reference. It is the Dell PowerEdge R210 II, with a 4-cores Intel Xeon E3-1200 series processor and 32 GiB of RAM.

The consumption of this server is measured up to 130 W at peak load and 44 W at idle[2]. This is slightly higher than the values announced by the manufacturer in its power and performance data sheet, but in accordance to other measurement on comparable servers [191]. The time required for turning on the server was not measured directly for this model. Very few works studied in detail the boot time of servers. There is a wide

[2]Power consumption using SiSoft Sandra benchmark, as measured by http://www.itpro.co.uk/635800/dell-poweredge-r210-ii-review.

Parameters	Meaning	Values
M	Number of machines	10
N_m	Processors per machine	4
R_m	Memory per machine	32 GiB
P_m^{idle}	Base machine consumption	44 W
P_m^{max}	Machine consumption at full load	130 W
C_m	Processor relative speedup	1
$P_m^{off \to on}$	Total power used during boot up	120 W
$t_m^{off \to on}$	Time for booting up	40 seconds
$P_m^{on \to off}$	Total power used during shut down	100 W
$t_m^{on \to off}$	Time for shutting down	15 seconds

Table 4.4 – Instantiation of the IT model described in section 4.1.2 (homogeneous data center, the values are given $\forall m$).

range of measured values depending the architecture and the software configuration [179], as shown for instance by Villebonnet et al. [191] measuring from 12 to 189 seconds. A value of 40 seconds seems reasonable for a headless server, as the measurements from [179] are comparable in the case of several machines with GNU/Linux distributions including a graphical session start. The measurements are even lower (around 30 seconds) when booting from network instead of using a hard drive for this purpose. To take into account the peak of power consumption which occurs when turning on and off a machine [147], we considered a consumption close to the full load one.

The simulated data center is composed of 10 of these servers, or 40 processors available. The model from section 4.1.2 is therefore instantiated with 10 identical machines with values summarized in table 4.4. This setup, resulting in a small infrastructure with a peak production of 1300 W, is chosen to be of similar scale with the GreenSlot [70] evaluation infrastructure (1765 W peak).

Electrical model and attractiveness The electrical infrastructure is composed of two electrical sources, as mentioned in section 4.2.1. The first one is a set of solar panels, providing a peak production of 1500 W. Depending of the photovoltaic module used, the cells technology, and the efficiency of the solar inverter, the required surface of solar panels may vary a lot. We will consider an efficiency of around 15%, which is a conservative value of what is commonly found in the current market, for modules using single or multi-crystalline technologies cells [37]. With such efficiency, a peak production of 1500 W is achievable with 10 m^2 of panels, assuming a peak solar irradiance of 1000 $W.m^{-2}$ [36]. The time window for prediction of energy availability and grid energy pricing, $T^{forecast}$ is set to 48 hours. As a matter of simplification, and as the electrical infrastructure is implemented in a simulator, the prediction model used here is an oracle, with perfect prediction during the whole window ($\hat{p}_i^{solar}(t') = p^{solar}(t'), \forall t' \in [t, t + T^{forecast}]$). The second source is the electrical grid, with an on-peak/off-peak pricing based on the values used in [70]. Two periods are considered for pricing: \$0.13/kWh from 9am to 11pm (on-peak), and \$0.08/kWh from 11pm to 9am (off-peak).

Having the maximum renewable sources production and the grid prices, we can define the missing constants used in electrical attractiveness functions. The values of h_{pos} and h_{neg} determine the rate at which attractiveness increase or decrease depending on the available power. We set both to a quarter of the peak renewable power, $h_{pos} = h_{neg} =$

375 W . For the variant B of the attractiveness function, we have the minimum and maximum grid energy price, $\theta_{min} = 0.08$ and $\theta_{max} = 0.13$. The *price factor* λ will be set by experiments detailed in section 4.3.1.

Scheduling algorithm Several parameters are used in the task placement algorithm itself (algorithm 4). Two couples of parameters, related to time steps (α_{step} and $step_{max}$) and to the time window considered for scheduling (α_{window} and $window_{max}$) are used by the algorithm, as described in section 4.1.4. A last parameter, α_{reboot}, control the aggressiveness of the shutting down of idle machines. The values used for the first four parameters, presented in table 4.5, were chosen based on preliminary experiments for giving good results for both short and very long tasks. However, they were not tuned specifically for the workloads used in this chapter and different values may lead to better results, at the cost of a longer execution time of the task scheduling algorithm.

For the parameters related to the time window, the chosen values allow the scheduling of tasks long after their due dates (up to 4 times the duration of the task with a maximum of 12h). This permits possible SLA violations if a heuristic considers it more interesting than using more energy at some point to respect the due date of a task. The ones related to the time steps are chosen to avoid too long steps for long tasks (the maximum time step is 30 minutes), while keeping steps long enough for short tasks (0.3 times its normalized execution time T_j).

Finally, choosing $\alpha_{reboot} = 2$ allows to keep a machine turned on if it is idle for less than twice the time needed to perform a complete reboot. Choosing higher value results in more idle machines not powered off and a lower value results in more aggressive reboot of machines to avoid any idle time. As the machines consume more power when shutting down or booting up than in idle state, this value may also be tuned with the characteristics of the machines used. For sake of completeness, the optimal in terms of power consumption and for an homogeneous data center is given by eq. (4.7). This equation gives the value of α_{reboot} for which the total energy consumed during a reboot cycle is equal to the energy consumed if the machine stayed idle.

$$\alpha_{reboot}^{optimal} = \frac{P_m^{off \to on} \cdot t_m^{off \to on} + P_m^{on \to off} \cdot t_m^{on \to off}}{P_m^{max} \cdot (t_m^{on \to off} + t_m^{off \to on})} \tag{4.7}$$

α_{step}	$step_{max}$	α_{window}	$window_{max}$	α_{reboot}
0.3	0.5 h	4	12 h	2

Table 4.5 – Values of parameters used in the scheduling algorithm.

Workloads The workloads are generated with the model described in section 4.2.1, which gives the values, for each task j, of the parameters S_j (submission date), D_j (its due date) and T_j (its execution time). Several values of flexibility factors *flexf* are used, depending on the experiments, and will be given along with the results. In order to keep the workload definition simple enough, all the tasks have the same CPU and memory requirements, $n_j = 1$ and $r_j = 1$ GiB. In our experiments, the measured average load of the computing infrastructure with such a workload is about 45 %, which is higher than most of the typical data centers [158], and similar to some modern cloud clusters.

To ensure the validity of our experiments, we generated a set of 10 different workloads for each flexibility factor used. Those 10 workloads are generated with the same probability

distributions, but using different random seeds. All the results given in the next sections were repeated on the 10 workload instances, and the values used are the average, unless specified otherwise.

4.2.4 Lower bound of grid consumption

The methodology presented here was focused on comparing our approach to others heuristics from the literature. However, it does not give hints to know how close those results are from the optimal. Scheduling problems are, unfortunately, known to be easily intractable, even for simple models [184]. In the particular problem presented in this chapter, we expect that obtaining an optimal solution (thanks to linear programming for instance) within a reasonable time can be achieved only on tiny instances. Therefore, we propose instead to study a lower bound of the energy consumed on the grid.

The main idea is to relax the problem in a continuous way, such that individual tasks are gathered into a fully parallelizable workload (described as a *mass*). This mass can be assigned indistinctly onto the machines, while keeping information about submission and due dates of the tasks. We keep the notations of the model presented in section 4.1.2 and rely on the following assumptions:

- A task may be executed on any number of processing unit (fully parallelizable tasks), suspended and migrated instantaneously with no cost.
- The data center is homogeneous (the subscript m, used in the notations from section 4.1.2, will refer to any machine and their speedup $C_m = 1$).
- The machines can be powered on and off instantly.
- The power consumed by a machine is proportional to the amount of processor used (each accounting for P_m^{max}/N_m).

As the problem has been relaxed with these assumptions, we should note that the lower bound would be unachievable.

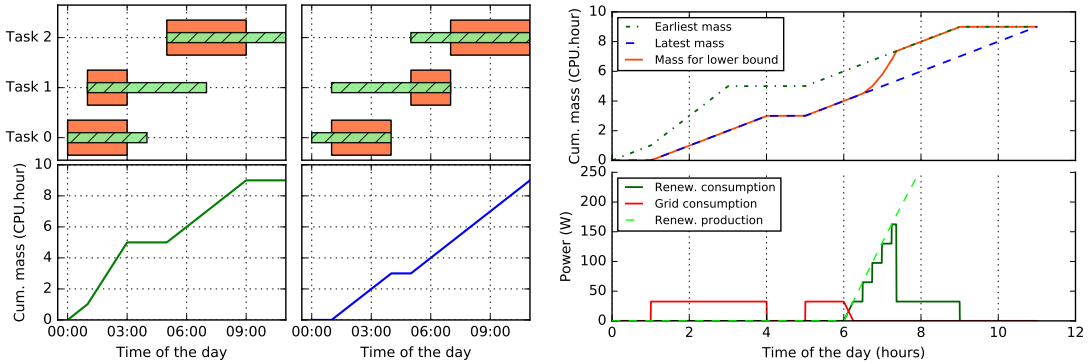
We focus on two characteristics of the workload. The first one, $Mass_{earliest}$, given by eq. (4.8c), is the cumulated mass if every task is scheduled as early as possible. It is easy to compute based on $Wl_{earliest}$, the mass given in eq. (4.8a). Equation (4.8d) gives $Mass_{latest}$, the second one, when the tasks are scheduled as late as possible while respecting their due dates, based on the mass from eq. (4.8b).

$$Wl_{earliest}(t) = \sum_j \begin{cases} n_j & \text{if } S_j \leq t \wedge S_j + T_j \geq t \\ 0 & \text{otherwise} \end{cases} \tag{4.8a}$$

$$Wl_{latest}(t) = \sum_j \begin{cases} n_j & \text{if } D_j - T_j \leq t \wedge D_j \geq t \\ 0 & \text{otherwise} \end{cases} \tag{4.8b}$$

$$Mass_{earliest}(t) = \int_0^t Wl_{earliest}(x)\ dx \tag{4.8c}$$

$$Mass_{latest}(t) = \int_0^t Wl_{latest}(x)\ dx \tag{4.8d}$$



(a) Tasks in workload and corresponding cumulated computing mass, when executed at earliest (left) and at latest (right).

(b) Solution of the lower bound method for this workload, with associated consumption accross time.

Figure 4.3 – Illustration of the method for finding the lower bound of grid energy consumption.

$$\forall t \quad 0 \leq Cpu(t) \leq N_m \cdot M \tag{4.8e}$$

$$Cpu(t) \geq Mass_{latest}(t) - Mass_{done}(t-1) \tag{4.8f}$$

$$Cpu(t) \leq Mass_{earliest}(t) - Mass_{done}(t-1) \tag{4.8g}$$

$$MassDone(t) = \int_0^t Cpu(x)\ dx \tag{4.8h}$$

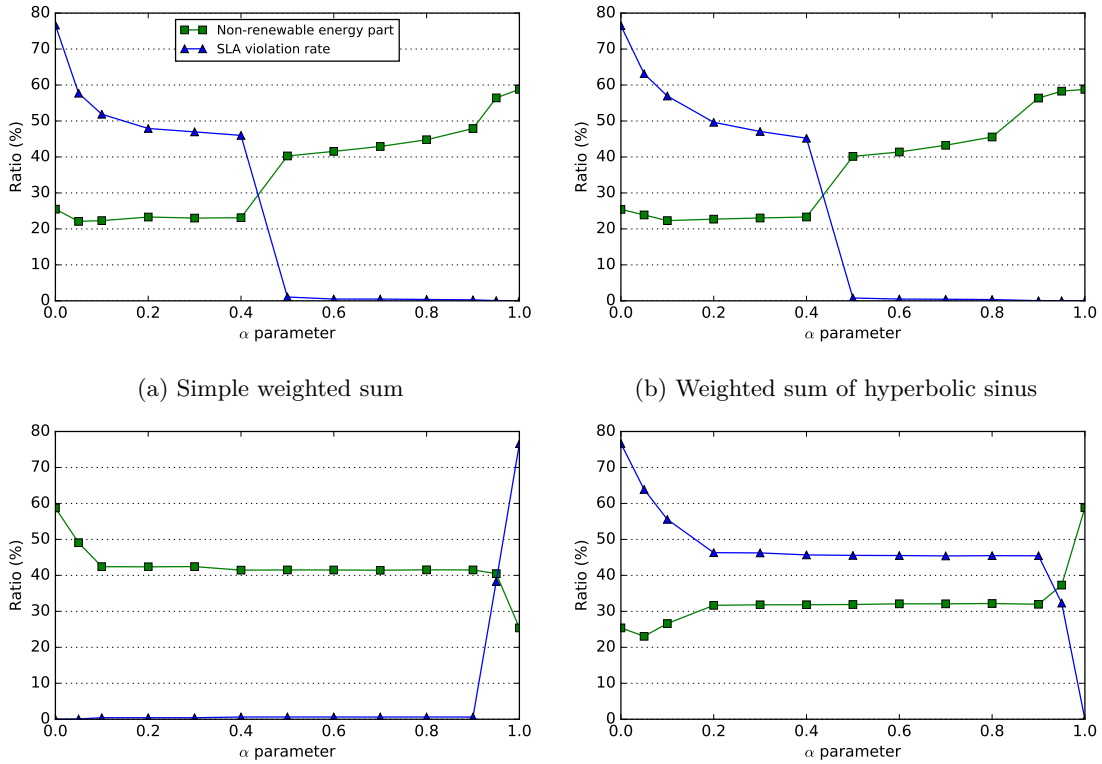
$$Cons(t) = Cpu(t) \cdot \frac{P_m^{max}}{N_m} \tag{4.8i}$$

$$Grid(t) = \max(0, Cons(t) - \sum_w^W available_w(t)) \tag{4.8j}$$

Figure 4.3a illustrates that using an example workload with only 3 tasks. The tasks are represented in the top figures, along with the period between their submission and their due dates (in green). The cumulated mass associated to the earliest and latest scheduling are given respectively in the bottom left and bottom right subfigures. Any valid scheduling which respects all the due dates must have, at each time, its cumulated mass between $Mass_{latest}$ and $Mass_{earliest}$. This is represented by the constraints in eqs. (4.8e) to (4.8h). $Cpu(t)$ is the total amount of processors used at a given time t, while $MassDone(t)$ is the associated cumulated mass.

Based on these constraints, it is possible to find a lower bound of the energy consumed from the electrical grid. Equation (4.8i) gives the consumption of the data center, $Cons(t)$. By removing the power produced by each of the W renewable sources, $Grid(t)$ represents only the power coming from the grid, given by eq. (4.8j). An illustration of the method, for the example workload, is shown in fig. 4.3b. At the top, the earliest and latest cumulated mass are represented, as well as the one found as the lower bound. The resulting consumption, for both grid and renewable, are given in the bottom subfigure.

In practice, we used a discrete time model, with time steps small enough, to compute the lower bound. The solution is therefore given by minimizing $\sum_t Grid(t)$. At each time step, the renewable power is used to do as much work as possible without exceeding the earliest cumulated mass. Then, if it was not enough to reach the latest cumulated mass, grid power is used for this purpose.



(a) Simple weighted sum

(b) Weighted sum of hyperbolic sinus

(c) Fuzzy-based with IT attractiveness as first objective

(d) Fuzzy-based with electrical attractiveness as first objective

Figure 4.4 – Impact of multi-objective functions parameter value (weighting or fuzzy margin) on SLA violation and on part of grid energy used.

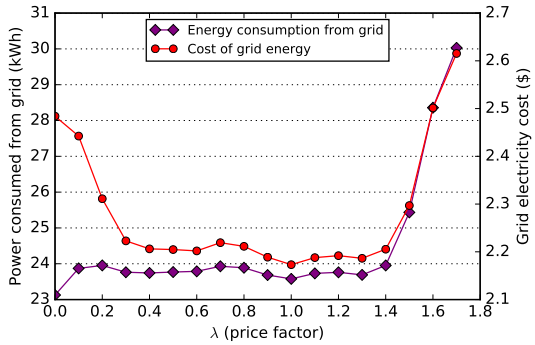
4.3 Results

In order to present our results, we will, in a first time, present the experiments we used to fix the parameters of the several multi-objective methods detailed in section 4.1.5, and the *price factor* used in variant B of the electrical attractiveness function. Then, using the values determined by those experiments, we will compare the performances of the different approaches and heuristics on several criteria, and for multiple *flexibility factors* of the workload.

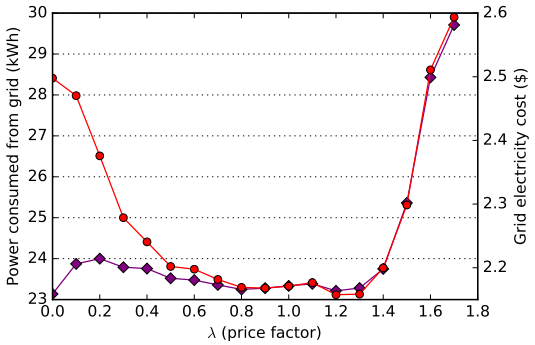
4.3.1 Choosing parameters

Multi-objective functions parameters We have a total of four different multi-objective methods, described in section 4.1.5, that may be used by ABBSH: the simple weighted sum, the weighted sum of hyperbolic sinus, the *fuzzy-based* with IT attractiveness as its first objective (fuzzy-IT) and the one with electrical attractiveness as its first objective (fuzzy-elec). For each of them, we have an α parameter which controls its behavior. For the two weighted sums, it is the weighting factor, and it is the fuzzy factor used to calculate the margin on the first objective for the fuzzy-based approach. The parameter β, used for the weighted hyperbolic sinus sums, was fixed to a value high enough to give more weight to the extreme attractiveness value, and we used $\beta = 2.5$.

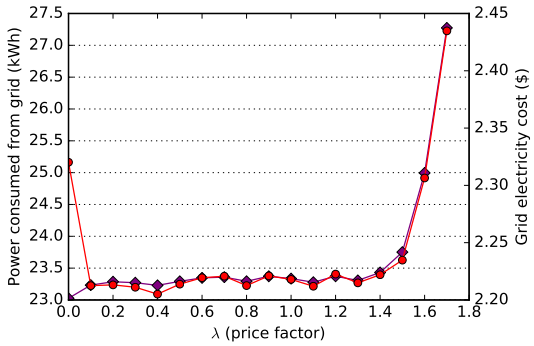
The value used impacts directly the preference toward either the IT or the electrical attractiveness. Ideal value depends on the formulation of attractiveness function. As these



(a) Simple weighted sum



(b) Weighted sum of hyperbolic sinus



(c) Fuzzy-based with IT attractiveness as first objective

Figure 4.5 – Impact of *price factor* variation on the amount of energy bought from the electrical grid and its total cost (with the variant B of electrical attractiveness function, which considers the price variations).

formulations do not change during the use of the scheduling algorithm, choosing the value is expected to be done offline and only once for configuring the scheduler.

To choose a value for them, we used workloads generated with a relatively high flexibility ($flexf = 8$), and used only the electrical attractiveness variant A. The experiments consist in changing the α value, and evaluating the impact of it on two criteria: the part of non-renewable energy used and the rate of tasks that finished their execution after their due dates. The results are shown in fig. 4.4. We are looking for a value of α, for each method, which minimizes both criteria.

For the weighted sums, the results in fig. 4.4a and fig. 4.4b are quite similar. The SLA violation rate decreases quickly when α becomes higher than 0.5, and the non-renewable part increase when α is close to 1. To keep a high QoS and a high renewable energy usage in the same time, we choose $\alpha = 0.55$ for both functions.

For the fuzzy-based approaches, a first look to fig. 4.4d indicates its worthlessness when the electrical attractiveness is the first objective. Indeed, the margin needed to have a SLA violation rate near zero seems to imply a less important renewable usage than for the other methods. For this reason, we will not keep it in the subsequent experiments. Figure 4.4c shows a very different result when the first objective is the IT attractiveness. In this case, we note that non-renewable energy usage decreases a little bit when α increases between 0.1 and 0.9, staying almost constant. Similarly, the SLA violation rate is slowly increasing, until it grows quickly when α reaches 0.9. We will choose a relatively high flexibility factor

of $\alpha = 0.75$, to keep the renewable energy usage as important as possible, and without risking to degrade too much the QoS.

Electrical attractiveness variant B With the multi-objective parameters fixed in the case of the variant A of the electrical attractiveness function (without taking the electricity price into account), we can study how to integrate the price in the variant B. The influence of the electricity price from the grid is controlled by λ, the *price factor*. The choice of it's value is to be considered as a decision of the data center operator. It fixes the trade-off between privileging the renewable energy (low value) or the cheap grid energy (high value). With $\lambda = 0$, the behavior is the same as with the variant A (we find $base = \alpha_{neg} = -0.7$), guaranteeing that the parameters chosen for the multi-objective methods will work as well for this case of the variant B. By increasing it, the attractiveness value, for a situation where renewable energy is not sufficient, will increase compared to the variant A by at most λ (in the case of the cheapest possible energy prices).

Figure 4.5 gives the results of non-renewable energy consumed (bought from the electrical grid) and its total cost. The impact of λ is similar for the different multi-objectives methods. The total cost decreases initially, but starts increasing again when the price factor becomes too high (≥ 1.3). This behavior is related to the total grid energy consumed. Indeed, the total cost is correlated with the energy consumed from the grid, and the average price for this energy. With small values of λ, the grid consumption increases a bit compared to $\lambda = 0$, but as the price is taken into consideration, the average price decreases in the same time. However, when λ becomes high enough, increasing it does not reduce significantly the average energy price, and the total price simply follows the grid consumption. Worse, when the price factor is too high, the electrical attractiveness function tends to favor the use of cheap grid energy instead of renewable one which covers only partially the requirement. Consequently, the amount of grid energy increases again and the total cost follows. Those high values are probably not the behavior wanted by any operator of data center with renewable energy sources, but shows how important it is to choose an appropriate value for λ.

With those results, we can choose a value for the price factor, used by the three different heuristics. We want a value that increases the least possible the non renewable energy consumption, while decreasing the grid energy cost the most possible. Therefore, setting the price factor $\lambda = 1.2$ seems reasonable, and will be used for the next experiments. With this value, compared to the variant A, the weighted sum with the variant B uses 2.2% more grid energy but reduces the cost by 12%. Similarly, with the weighted sum of hyperbolic sinus, the consumption from the grid is 0.7% higher for a cost saving of 13%, and respectively 1.0% and 4.9% for fuzzy-IT.

4.3.2 Impact of the workload flexibility

With the multi-objective functions parameters set and electrical attractiveness well defined, we can compare the performance of the different heuristics depending on the flexibility factor used to generate the workload. Greater flexibility factor gives tasks with more time between their submission and their due dates and therefore more opportunity for a scheduling algorithm to delay them to a better time while respecting their SLA. We used 6 different values for the flexibility factor *flexf*: 0 (resulting in tasks needing to be executed immediately after their submission), 1, 2, 4, 8 and 16.

For each value of *flexf*, 10 workloads were generated, using the same parameters described in section 4.2.1 but with different random seeds. We evaluated 9 different heuristics which are the following.

- the 3 GreenSlot variants (original, with partial modifications and fully modified)
- ABBSH with electrical attractiveness function A in combination with 3 multi-objective methods (weighted sum, weighted sum of hyperbolic sinuses and fuzzy-based method with IT attractiveness as first objective)
- ABBSH with electrical attractiveness function B in combination with the same 3 multi-objective methods

These 9 heuristics were each applied to all of the 10 workloads of each value of *flexf*. Several metrics were measured (percentage of tasks for which SLA was violated, amount of grid energy consumed and total price of this grid energy) to compare the different heuristics. During the rest of the section and unless stated otherwise, the results presented are the average value over the execution with the 10 workloads for a given combination of heuristic and flexibility factor.

GreenSlot versions Figure 4.6 gives the results for the three GreenSlot versions. We note the high SLA violation rate for the *original* version, in fig. 4.6a, which reaches 98% when the workload flexibility factor is 0. The violation rate is less than 2.5% for higher flexibility values, but is still several times more important than with the two other versions, which include our modifications. The same figure shows that the violation rate for the *partially modified* and *modified* versions is increasing with the flexibility factor.

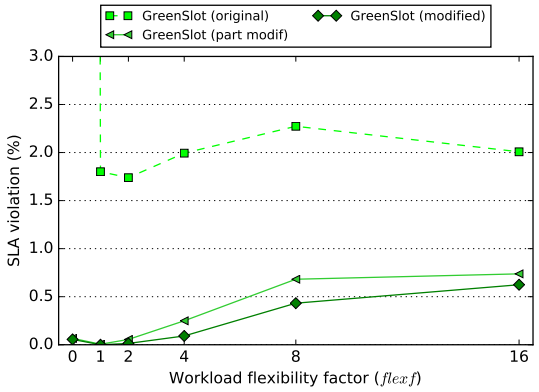
Figure 4.6b presents the power consumption from the grid for the different GreenSlot versions. Each tends to consume less grid energy when the flexibility factor increases. The *original* version consumes always more than the two others, for the reasons given in section 4.2.2. The cost of the grid energy, given in fig. 4.6c, is very similar to the amount of energy drawn from the grid. However, we can notably note that the cost is not exactly proportional to the energy consumed. The results suggest that GreenSlot scheduler use cheaper energy when the flexibility increases, as we can expect from a price-aware algorithm.

We can see how our modifications allow to improve the performances of the GreenSlot approach when used with our kind of workload, composed in large proportion of short tasks. As the *modified* version is better in all criteria compared to the others, we will use it as a reference when presenting the heuristics of our approach.

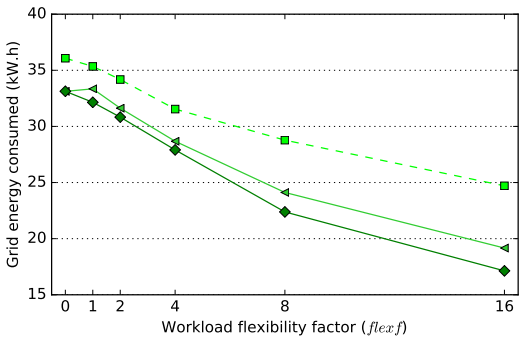
Electrical attractiveness variant A Figure 4.7 presents the results obtained with the several multi-objective methods of ABBSH, using the variant A of the electrical attractiveness function (no grid energy price consideration) and the GreenSlot *modified* version. The SLA violation rate (fig. 4.7a) has a similar shape for all the heuristics, increasing with the flexibility factor. Our approach leads to a bit more violations compared to GreenSlot, especially when the flexibility factor is high.

The consumption of non-renewable energy, shown in fig. 4.7b, is decreasing when the flexibility factor increases for all the heuristics. The fuzzy-IT method gives better results than the others when the flexibility factor is low (≤ 4 at least) or high (16), but is worst for medium value. Both simple weighted sum and the weighted sum of hyperbolic sinus give almost identical results, always equal or worst than the modified GreenSlot.

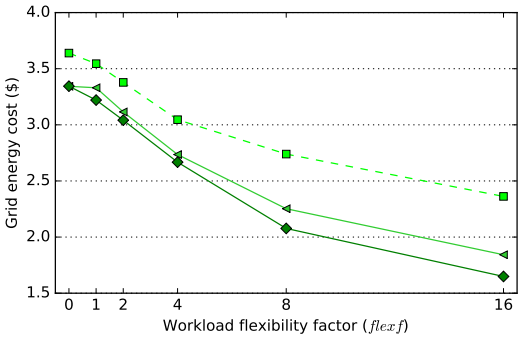
As the total cost for the grid energy is correlated with the amount of grid energy used, the behavior of the different heuristics is similar for this metric (fig. 4.7c). We can note, however, that GreenSlot takes advantage of the tasks flexibility to use cheap energy, which allows to reduce the cost more efficiently with high flexibility values than our approach with the variant A of electrical attractiveness. In addition, the fuzzy-IT method of our approach seems to use cheaper energy than the two others, without the grid price being



(a) SLA violation rate. The data point for GreenSlot original with a flexibility factor of 0 is not visible with this scale (its value is 98%).



(b) Energy consumption from the grid.



(c) Cost of the grid energy.

Figure 4.6 – Impact of flexibility factor over several metrics, for different versions of the GreenSlot scheduler implemented.

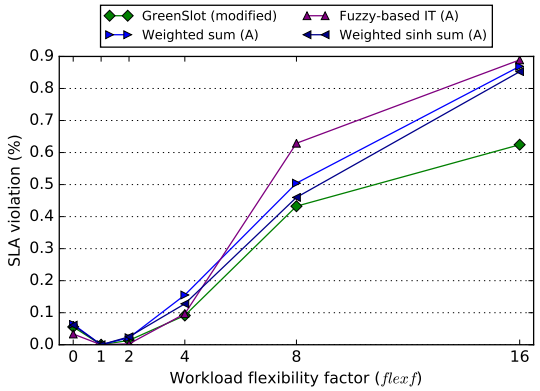
considered by the heuristic.

Electrical attractiveness variant B The same metrics are presented, for the variant B of our electrical attractiveness function, in fig. 4.8. SLA violation rate, in fig. 4.8a, follows a similar law than with the other cases, tending to increase along with the tasks flexibility. For low flexibility factors, the two weighted sum methods give much more violations than the fuzzy-IT one, or the modified GreenSlot algorithm.

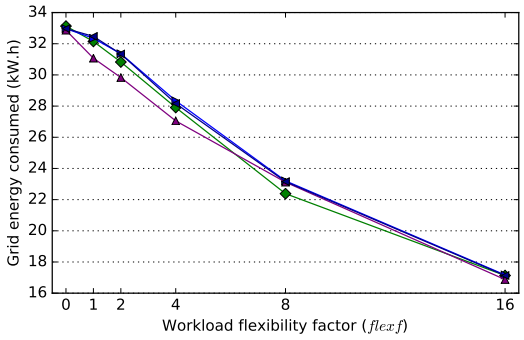
As we take into account the grid energy price with this electrical attractiveness function, we note that the grid energy consumption (fig. 4.8b) and cost of this energy (fig. 4.8c) are much more similar than for the variant A.

4.3.3 Comparison between approaches

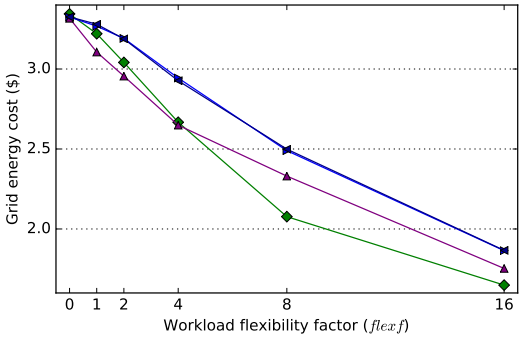
Figure 4.9 summarizes the results for most of the presented approaches and heuristics. The results of ABBSH using the weighted sum of hyperbolic sinus as multi-objective methods are not shown as they differ only slightly from using the simple weighted sum. Three flexibility factors are considered, 2, 8 and 16, which seem representative of workloads with respectively a low, medium and high level of flexibility. In this figure, the results obtained by using a first-fit scheduler are also represented, illustrating a traditional scheduling algorithm with no renewable energy consideration.



(a) SLA violation rate (percentage of the total task number)



(b) Energy consumption from the grid.



(c) Cost of the grid energy.

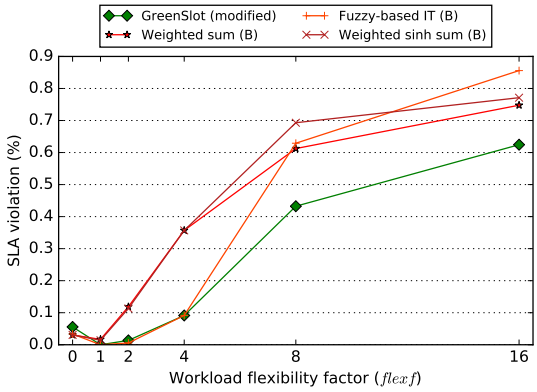
Figure 4.7 – Impact of flexibility factor over several metrics, for ABBSH with different multi-objective functions, using variant A of electrical attractiveness function (without consideration of grid energy price).

For workloads with low flexibility, our approach using the fuzzy-based method outperforms the GreenSlot one, even with all our modifications. Whereas, for *flexf* = 2, the GreenSlot *modified* version reduces the use of grid energy by 7.6% compared to a first-fit scheduling, our approach, with the fuzzy-IT method and the variant B of electrical attractiveness function, reaches 10.5%. The cost saving is even more important, with only 9.6% for GreenSlot compared to 12.4% for fuzzy-IT.

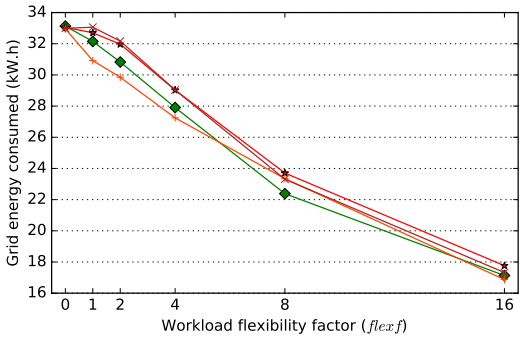
The same heuristic (fuzzy-IT with variant B) gives results similar to the *modified* version of GreenSlot for the workloads with high flexibility. The GreenSlot scheduler reaches 48.2% grid energy saving and 51% cost saving. Our ABBSH algorithm performs slightly better on this scenario, with a grid energy saving of 49.4% and a reduction of 51.2% of the cost, compared to the traditional scheduler.

Our approach is slightly outperformed for workloads with medium flexibility (*flexf* = 8). In this case, our best heuristic for reducing non-renewable energy use is the fuzzy-IT with variant A, reaching 31% saving, and the best one for reducing cost, with 35.4% saving, is the simple weighted sum with variant B. The GreenSlot *modified* version reduces the grid consumption by 33.2% and the cost by 38.6%.

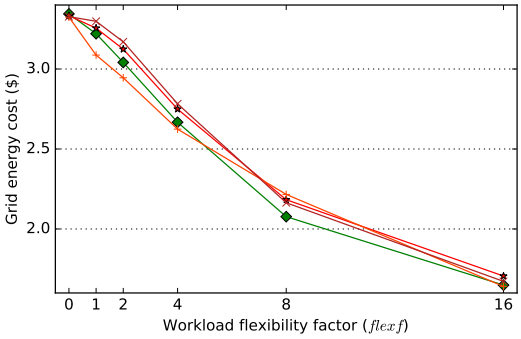
In all the scenarios, the *original* version of GreenSlot gives the worst results, showing again that it is not adapted to the kind of workload we used. Notably, when the flexibility factor is low, it actually increases both non-renewable energy use (+2.5% for *flexf* = 2) and total cost (+0.4%) compared to a first-fit scheduling. In addition, it causes important



(a) SLA violation rate (percentage of the total task number)



(b) Energy consumption from the grid.



(c) Cost of the grid energy.

Figure 4.8 – Impact of flexibility factor over several metrics, for ABBSH with different multi-objective functions, using variant B of electrical attractiveness function (consideration of the variation of the grid energy price).

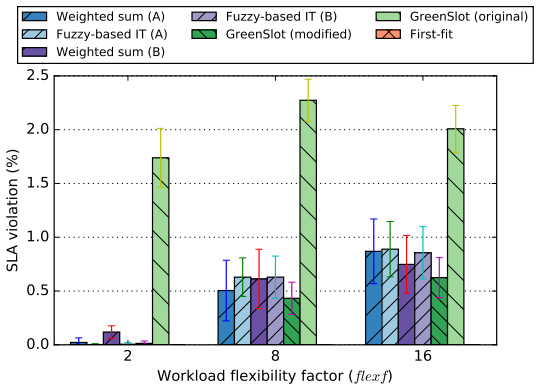
SLA violation rate compared to the other heuristics, as shown in fig. 4.9a.

The other heuristics give an average SLA violation rate lower than 1% on each of the three scenarios. However, at the exception of case of workloads with low flexibility, the violation rate of our heuristics is slightly higher than the one of the GreenSlot version with our modifications.

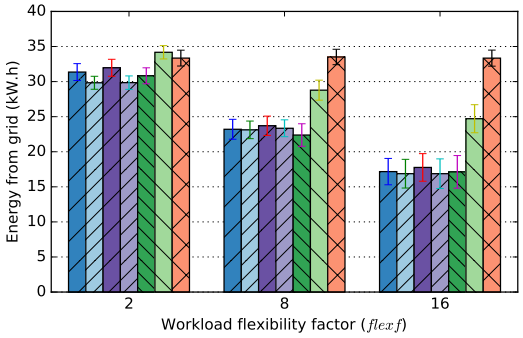
4.3.4 Comparison to the lower bound

Based on the method presented in section 4.2.4, we computed a lower bound of the grid energy consumption for every workload used in the previous experiments. The results for different values of the flexibility factor are given in fig. 4.10a, along with one of the ABBSH heuristic and the two variants of GreenSlot. To interpret those results, it is important to emphasize that the model used for this lower bound does not take into account some of the energy consumed normally. Notably, the energy needed to power on and off the machines is not accounted.

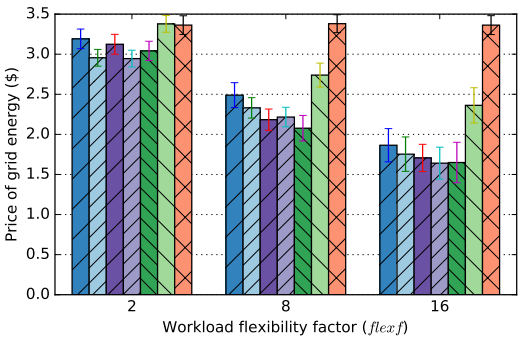
Therefore, the total energy consumed (both from grid and renewable sources) is significantly less than with the tested heuristics. For all of them, this amount of total energy varies little depending on the flexibility. The average, for each of the variants of our approach as well as for the modified GreenSlot algorithm, are between 55.7 kWh and 56.6 kWh. This value is as high as 60.5 kWh for the original GreenSlot variant. By con-



(a) SLA violation rate (percentage of the total task number). First-fit is always at 0, and therefore not visible.



(b) Energy consumption from the grid.



(c) Cost of the grid energy.

Figure 4.9 – Comparison of the results between the approaches and heuristics, for workloads with different flexibility values. The standard deviation is represented, as each experiment is repeated with 10 workloads generated using different random seeds.

trast, the solutions from the lower bound method give an energy consumption of 50.4 kWh.

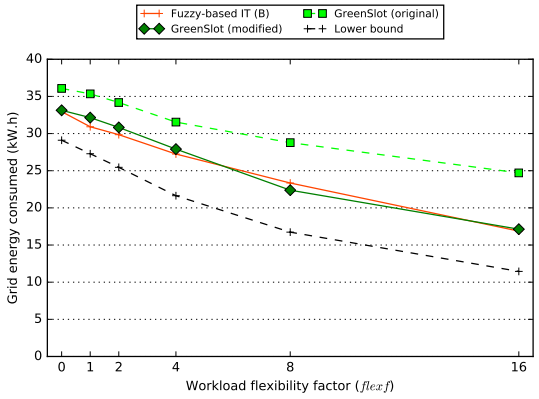
The shape of the grid energy consumed curves, from fig. 4.10a, are similar for the different heuristics and for the lower bound. The absolute difference between the lower bound and the best result of all the heuristics increases with the flexibility factor for small values (from 3.8 kWh for $flexf = 0$ to 5.4 kWh for $flexf = 4$). For higher flexibility factors, the difference is almost constant (5.6 kWh and 5.4 kWh respectively for $flexf = 8$ and $flexf = 16$).

Figure 4.10b shows the percentage of the energy consumed provided by the renewable sources. The method used to compute the lower bound of the grid consumption does not give an upper bound to the percentage of renewable energy used. However, those values help to take into account the differences in total energy consumption.

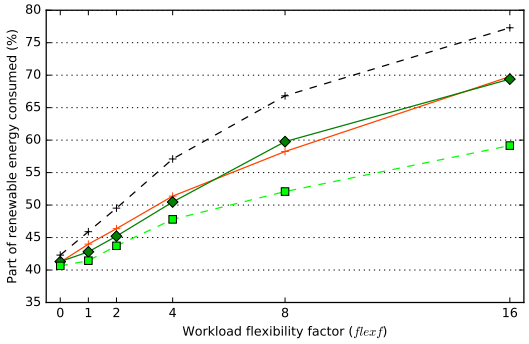
The lower bound shown in figs. 4.10a and 4.10b, demonstrates that the heuristic we propose provides gain of the same order of magnitude of what could possibly be reachable under our hypothesis.

4.4 Discussion

One of the interesting findings of our experiments concerns the impact, on the performances of the tested heuristics, of workload flexibility (e.g. the average time available for



(a) Energy consumption from the grid.



(b) Portion of renewable energy in the total consumption.

Figure 4.10 – Comparison of the results between different approaches and the lower bound of grid consumption.

scheduling each task and meeting their due dates). While the increase of flexibility allows, as expected, to improve the results in terms of non-renewable energy and cost saving, it also leads to an increase of SLA violations. It may seem counter-intuitive, but is easily explained by the increased usage of computing resources when the energy is available. As some tasks, with important flexibility, are delayed to be executed at those moments, a task with low flexibility submitted when all machines are already reserved is likely to be scheduled after its due date. It may be possible to overcome this issue by using a heuristic which takes into account a prediction of tasks submitted in the future, in order to keep some computing resources available for them.

The relationship between the workload flexibility and the reduction of the cost and of the grid energy use seems quite complex. Beyond a certain flexibility ($flexf > 8$ in our experiments), increasing it still allows to give better results, but proportionally less important to what is achievable by increasing it as much below this threshold. This finding is reinforced by the results of our lower bound of grid consumption. The saving of grid energy as a function of the flexibility factor behave almost linearly for $flexf \in [0, 4]$, as shown in fig. 4.10. However, we believe that it is particularly related to the profile of the renewable source we used. Indeed, our simulated solar panels give a very regular power, as if every day was perfectly sunny. If, instead, we had simulated a more realistic weather, a cloudy day would involve to move as much tasks as possible to the next day, and therefore would probably take advantage of tasks with very high flexibility.

Contrary to other studies, like [70, 71], our work shows how the potential energy saving of a renewable-aware scheduler is constrained by the SLA negotiated with the users. Also, we believe it is important to define a metric which gives the amount of freedom for the scheduling of a workload. The definition of this metric would depend on the kind of workload considered. In the case of a batch tasks with due dates, a more generic equivalent of our *flexibility factor* may be used (for example a measurement of the average or median of the flexibility of the tasks). We see two major reasons for defining such metrics. Firstly it would make possible to compare more easily the performances of different approaches, and to evaluate a given approach with several well identified hypothesis of SLA level. Secondly it would allow to measure it in real data centers, either to predict the performances of a given approach for this specific data center, or to use this measurement as a reference for future researches.

The aforementioned relationship between SLA and potential energy saving could also

be used to study possible *green pricing* for data center operators. The operators can promote lower SLA level to the users, by the way of incentive schemes, in order to save energy and money. Such incentives may concern the price of the jobs, but not necessarily limited to that. Indeed, other parameters also play a role in the energy saving behavior of individuals [65].

The results of the several variants of our ABBSH algorithm show that all of them reduce both grid energy usage and energy cost compared to a traditional scheduler. Taking the grid energy price into account (with the electrical attractiveness variant B) leads to a small increase of non-renewable energy use, but allows an important reduction of the grid energy cost, as illustrated by figs. 4.9b and 4.9c. The *fuzzy-based* multi-objective method provides, in overall, the most interesting trade-off between respect of SLA and energy or cost saving.

By introducing some modifications into the GreenSlot scheduler [70], we adapted it to the kind of workload we used for validating our approach, as evidenced by the results from fig. 4.6. In the original article, the authors used a very grid-oriented scientific workload. In contrast, the workload we used, based on the traces of a Google cloud cluster, is more representative of a typical batch workload executed in cloud infrastructures, like map-reduce jobs or periodic indexing tasks. With our modifications, the performances of GreenSlot are quite similar with those of ABBSH. Whereas GreenSlot is a bit more efficient than our heuristics with a medium flexibility, our approach gives slightly better results for workloads with low flexibility ($\mathit{flexf} \leq 4$).

A notable fact is the proximity of the results of both approaches. For every flexibility factor value, and for both grid energy consumption and cost, the results of our best heuristic and those of the modified version of GreenSlot are included into a 5% interval. We believe that such close results for two different heuristic-based approaches suggest they are near to a local optimum of this kind of online and greedy heuristics. The results of the proposed lower bound gives only a partial answer. Looking to the absolute difference of the grid consumption with the tested heuristics and to the percentage of renewable energy used, it seems possible to slightly improve our approach. Unfortunately, we are currently unable to tell exactly how much improvement can be done, as this lower bound does not take some of the characteristics of the problem into account (such as the cost of powering on and off the machines). Trying to solve the formal optimization problem, at least for small instances, would allow to identify exactly the improvement margin we can expect.

4.5 Conclusion

In this chapter, we presented ABBSH, a new algorithm with several variants for scheduling batch tasks with awareness of renewable energy and electricity cost. Our scheduler, contrary to other approaches, never deals directly with the model of the electrical infrastructure, but instead uses partial and abstracted information provided by a black-box function. By using these hints to guide the scheduler, our approach gives results close to, or slightly better, to a reference scheduler of the literature which has a full knowledge of the electrical model.

Experimental results also give many insights on the relationship between the magnitude of the SLA negotiated for the tasks of the workload, and the performances of the different heuristics studied. With SLA allowing more freedom on the time a submitted task should be executed, a renewable-aware scheduler can take more efficiently advantage of the intermittent nature of most renewable energy sources. As an indirect consequence, this increases the part of the computing resources used when the energy is the more available,

leading to an inability to satisfy stricter SLAs of tasks submitted at these times.

The performances of the different heuristics are also compared to a lower bound of the grid energy consumption. The results show that our approach performs already well and gives an idea of maximum margin of improvement. In addition, such lower bound formulation with lower complexity than the real optimization problem can be used to continue to investigate on the relationship between SLA and possible cost and energy savings.

With results being comparable to a scheduler with total knowledge of the electrical model, this approach shows that it is possible to use efficiently the renewable energy in a data center with separated (electrical and IT) management systems. This open the perspective of changing the electrical infrastructure, by adding electrical sources, storage or changing its objectives without being tied to the model handled by a given scheduling algorithm. However, with the presented approach, both systems simply *cooperate* with the other, and the scheduling algorithm is not designed to minimize the amount of communication needed to find a solution. Moreover this approach is still tied to the underlying computing resource and workload models. Finally, the unavoidable uncertainties in renewable power prediction are not taken into account and the impact of prediction errors is not studied.

While the uncertainty is explored later in chapter 6, the other points are addressed by the contribution presented in the next chapter. It extends the concept of *cooperation* between the electrical and the IT infrastructures, by considering each as a totally independent black-box and exploring the possible trade-offs that satisfy them.

Chapter 5

Multi-objective, cooperative power planning

The greatest challenge to any thinker is stating the problem in a way that will allow a solution.

— Bertrand Russell

Contents

In the previous chapter, we studied the management of on-site renewable energy sources (RES) variability and intermittency as an IT resources scheduling problem. While it still considers a single datacenter powered with on-site RES, the work presented in this chapter focuses less on the computing part, which is considered as a black-box. The overall problem is therefore tackled as a multi-objective power planning problem, in which the computing infrastructure (machines and allocation of their resources to the workload) is as important as the optimization of the electrical sources and storage devices. These two parts of the overall infrastructure, IT and electrical ones, are each optimized independently under power planning constraint and the details of their model and optimization method is hidden (black-box model).

The multi-objective aspect of datacenter management with RES has already been highlighted. At least two major objectives are easily identified: minimizing the ecological footprint and maximizing the operation of the datacenter itself, in terms of quality of service and/or economical revenues. Considering that the two concerns are hardly miscible, transforming the problem into a single-objective one using weighted sum or ε-constraint appears insufficient. Hence we propose to tackle the bi-objective problem directly, trying to reach the optimal Pareto front representing the ideal trade-offs. With this multi-objective approach, we aim at providing a set of solutions that offers datacenter operators

the freedom to choose the right trade-off between carbon footprint and immediate income according to their green commitment.

We consider an existing datacenter powered partially by renewable sources and potentially any energy storage devices (ESD), encompassing the design proposed by the ANR Datazero project. The electrical part, or Power Decision Module (PDM), is in charge of planning the electrical sources commitment, according to the green objective. The Information Technology Decision Module (ITDM) is responsible for scheduling the tasks and managing the power state of the machines, having the economical income (through the quality of service) as objective. Some of the practical and theoretical reasons of using black-boxes are detailed in section 5.1.

In this model, each black-box is able to associate a value to any planning of power. This value reflects the impact of the given planning (power available or power to produce) on its own objective, if it is applied during the upcoming time window.

By using these two black-boxes, an overall optimization module explores possible solutions, trying to find a set of power plannings which covers a wide range of trade-offs between the two objectives. As they rely on complex models, the valuations of the power planning by the black-boxes are expected to be costly, therefore we aim at finding reasonable trade-offs using a limited budget of solution evaluation requests.

Several contributions are presented along this chapter.

- we design USPEA2, a multi-objective evolutionary algorithm integrating well with objective approximation techniques
- a novel method for online approximation of black-box functions defined on time series is proposed, using Haar transform to perform dimensionalty reduction
- an extensive performance evaluation is performed over several scenarios of renewable production based on realistic data from the National Renewable Energy Laboratory (NREL) [144], using a simplified infrastructure model
- the approach is validated using a datacenter model and scheduling algorithm from the literature [27], along with a workload based on the Google cluster traces

A description of the overall infrastructure and of the problem addressed by our approach are detailed in section 5.1, followed by a brief description of existing methods and concepts used in this context in section 5.2. Our multi-objective optimization approach, USPEA2, is presented in section 5.3 along with the two surrogate methods for time series defined objective functions. Section 5.4 covers the methodology of the experiments and describes the simplified model used for extensive evaluation. The results of the experiments are split between section 5.5 for preliminary results used for tuning some parameters and section 5.6 for the core results. They are interpreted and discussed along section 5.7 and finally section 5.8 concludes this chapter.

5.1 Problem statement

The optimal management of computing resources and workload with variable power constraint is known to be a hard problem. The same goes for the optimal engagement of multiple electrical sources and storage devices to maintain a stable power supply. Despite the high complexity of these optimization problems, a lot of works proposed more reasonable heuristics which give good enough results in practice, for either the computing or the electrical sources management. The biggest challenge to provide efficient solutions to the joint optimization probably lies directly in the complexity of both problems.

Instead of proposing new methods to solve the overall problem, the proposed approach keeps the two independent sub-problems, which can be implemented using existing meth-

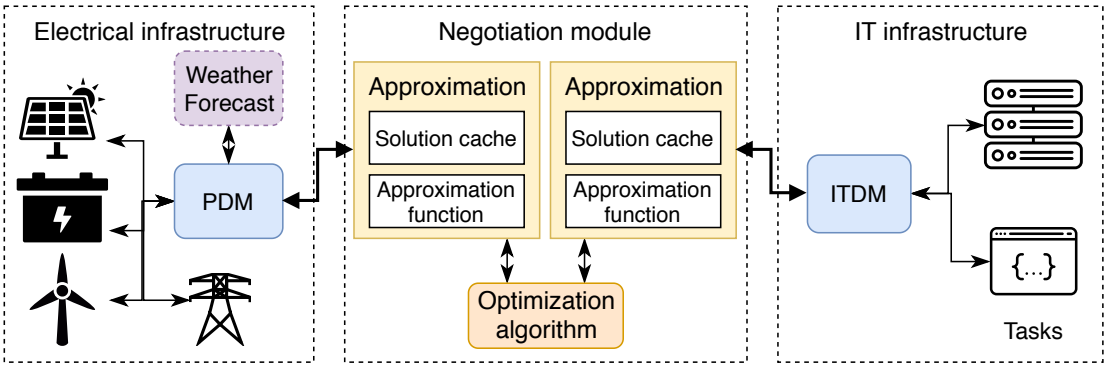


Figure 5.1 – Overview of the datacenter and renewable sources infrastructure, using the proposed multi-objective optimization approach. The two infrastructures are represented along with their decision module, Power Decision Module (PDM) and IT Decision Module (ITDM).

ods with minimal modifications. We propose a generic negotiation algorithm, which is in charge of driving the two distinct optimizations in order to find solutions that satisfy both.

An overview of the whole datacenter infrastructure is shown in fig. 5.1. Each *decision module* represents both the optimization algorithm, internal model and data for the electrical part or the computing part of the infrastructure. The decision modules are black boxes for the negotiation module. These modules are assumed to be designed and developed by experts in their own field, either electrical engineering, or datacenter management. Their purpose is to evaluate possible power consumption or production plans represented by *power profiles*, which are sequences of power values at each time step in a given time window. Such evaluation is done by an internal optimization, leading to associate to a power profile a *utility* value as an indicator of preference.

In a power profile, the set of possible values at each time step is noted $\mathcal{W} \subseteq \mathbb{R}$. A power profile for a window of T time steps is therefore an element of $\mathcal{W}^T$. Utility value may be any number in $\mathbb{R}$ with the only assumption that preferred solutions lead to greater values. We can abstract the evaluation of the power profiles by one of the decision modules as a function $f : \mathcal{W}^T \to \mathbb{R}$. This utility evaluation function is noted f_{itdm} for the IT decision module and f_{pdm} for the electrical one.

The generic optimization problem here consists therefore of finding a power profile p in $\mathcal{W}^T$ such as the associated utilities $f_{itdm}(p)$ and $f_{pdm}(p)$ are both as good as possible. Each utility value represents however a distinct objective. Furthermore these objectives are expected to be antagonists in most cases, as consuming more energy should favor the computing part at the expense of the electrical part.

The whole problem should therefore be considered as a multi-objective one. Without additional information on the relative importance of each objective, solutions may only be compared by the way of Pareto dominance as presented in section 2.1. By knowing the Pareto front of a multi-objective problem, a decision maker has as much information as possible to select one of these optimal trade-offs. We will not discuss here on the way this final decision is taken, but instead on providing the necessary information. For this purpose, having the complete and exact Pareto front is ideal, but not always feasible. When not feasible easily, as in the problem presented here, finding a good approximation in reasonable time is already valuable.

The optimization problems solved by each decision module are complex and we expect

good quality heuristics with realistic models to be both resource- and time-consuming. Consequently, the cost of evaluating the utility associated to a power profile is far from being negligible. The time available to take a decision is however limited, at least because this decision concerns an upcoming time window and becomes useless if taken too late. To take this into account, the number of possible evaluations of power profiles is constrained by a budget, denoted $eval_{max}$. Without more prior knowledge of the time required for PDM and ITDM, the two are considered in the same budget, such as the evaluation of a solution by the two decision modules counts as 2 costly evaluations. The complete problem may therefore be formulated as: find the best Pareto front approximation set $\mathcal{A}$ by using at most $eval_{max}$ power profile evaluations (represented by either f_{itdm} or f_{pdm}). This budget constraint is one of the characteristics of approaches dealing with optimization of expensive black-boxes, such as Regis [160].

To give an idea of the time requirement for a given budget, simple hypothesis are proposed. We suppose first that the total time is largely dominated by the costly function evaluation and that a given profile is evaluated by both the ITDM and the PDM simultaneously, as if each decision module optimization process is running on a different machine. With a given budget $eval_{max}$ and an average execution time of the slowest decision module of d, the total duration is $eval_{max} * d/2$. Therefore, with a budget and $eval_{max} = 100$ evaluations and a costly evaluation duration of $d = 1$minute, the total duration is close to 1 hour (50 minutes). For this reason it is critical to keep the budget in a smaller range compared to some other multi-objective evolutionary algorithm (MOEA) applications where the optimization is totally off-line. The target budget here is between 100 and 400 evaluations, which may both be interesting values depending the duration of a single costly evaluation and the time horizon of the optimization.

5.2 Multi-objective optimization of expensive black-boxes

Finding an approximation of the Pareto front for a multi-objective problem is well studied in the field of MOEA. The core of evolutionary algorithms is the exploration of the solution space by many creation and modification of individuals (representing possible solution) and the evaluation of their objective values. Thousands of possible solutions are usually evaluated by these approaches. Hence, when the evaluation of a single solution is computationally expensive or the time available is limited, this amount of evaluation is prohibitive.

To reduce the time spent evaluating solutions by evolutionary algorithms, it is common to use an *approximation function* (or surrogate function) to substitute some of the evaluations by cheap approximations [22, 93].

5.2.1 Integration of MOEA with surrogate models

The integration of surrogate models with MOEA may be done in different ways, with three main categories in the existing works. A first design, referred here as *synchronous* integration, consists of applying the chosen MOEA using the real objective functions for a small number of iterations, followed by its application with the surrogate objective functions for more iterations [19, 138]. This process is repeated multiple times, using the last solution set as initial population for the next phase and optionally updating the surrogate model after each non-surrogate phase. Such approaches do not require modifications of the MOEA used internally and can use surrogate method with relatively costly construction cost, but are easily impacted by a surrogate performing badly during

a single phase and not well adapted for a low budget of costly objective evaluations (few hundreds).

The second category, *asynchronous* surrogate integration, mixes evaluated and surrogate-approximated individuals in the same MOEA population, performing a costly evaluation only if an approximated individual is expected to improve either the solution set or the surrogate model [98]. A few modifications in existing MOEA are usually required, or completely novel approaches are proposed, but such methods may be applied even with low evaluation budget and the continuous verification of potentially good solutions reduces the impact of badly approximated individuals.

In addition to the synchronous or asynchronous integration, a few works also tried to integrate the surrogate model more deeply into the population selection and evolution process [3]. In this case, the main goal of the surrogate model is to allow the discovery of interesting areas of the solution space, either expected to improve the overall solution quality or to explore unknown parts of the space. Contrary to the two other categories, the surrogate model is not used simply as a substitute for evaluating possible solutions, but to perform local or global search in the solution space. Some solutions located in the area of interest are then evaluated using the costly objective function, which in turn improves the knowledge of the surrogate model.

We propose in this chapter a new method, which belongs to the *asynchronous* category and adapted to low evaluation budgets. Contrary to [98], the individuals are evaluated based on the number of generations during which they survive as good solutions. In addition, it embeds a mechanism to perform a costly evaluation when surrogate model considers to lack local information to accurately approximate a solution.

5.2.2 Surrogate methods for black-boxes

A surrogate of a costly function $f : \mathcal{S} \to \mathbb{R}$ is a function $f' : \mathcal{S} \to \mathbb{R}$ which gives, for each input $s \in \mathcal{S}$ a value as close as possible of $f(s)$ with lower cost (in terms of time or memory for instance). When the analytical form of f is available or when an algorithm computing values of f is known, is may be possible to formulate a simplified version keeping the main features, hence providing a surrogate f'.

When f is a black-box, such methods are not feasible. Assuming the function is not totally arbitrary, for instance by exhibiting some local linearity or continuity, regression techniques may be used after sampling some of the solution space. Building a surrogate by sampling the original function f is sometimes feasible offline, before the surrogate is used. This is useful when the surrogate is used several times for different purpose, allowing an expensive but precise sampling of f beforehand.

We focus here on *online* surrogate building instead, because of the definition of our problem. Each decision module is a costly black-box function for which a surrogate is desired. It is expected that their implementation depends on both environmental factors (weather forecast or planned workload submission) and internal state (current state of charge of the batteries or running tasks). For these reasons, building a surrogate model of these functions offline, in a static way, is not adapted. Online methods may be used instead, which are progressively learning the shape of the utility functions as new solutions are evaluated. In addition, because the solution space consists in time series of T time steps, with up to hundreds of time steps considered, surrogate methods should handle many-dimensions inputs.

Common surrogate methods used in evolutionary algorithms include Kriging models, artificial neural networks and Radial Basis Functions (RBF) approximation [22, 93, 170, 172]. Some of them, such as Kriging, are known to be prohibitively costly for many-

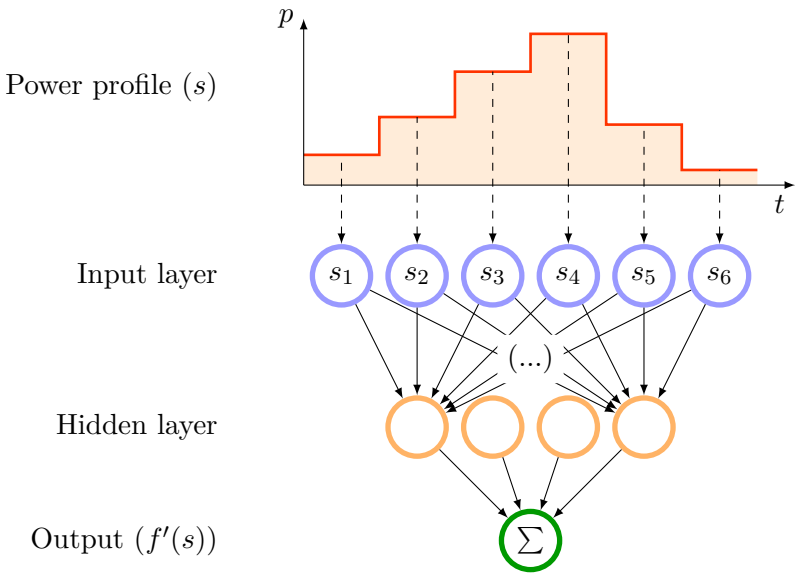


Figure 5.2 – Representation of a multilayer perceptrons neural network used as a surrogate of ITDM or PDM objectives, with a single hidden layer.

dimensions problem spaces [172]. Two methods from the literature, which are known to scale better with the number of dimensions, are considered in this chapter: multilayer perceptrons (a form of artificial neural network) and RBF. We believe that these methods, while performing well in many problems, are not well adapted for simultaneously having large solution space and few known samples caused by the low evaluation budget we are targeting. Two new methods, named Average Distance (AD) and Multiresolution Haar Transform (MHT) are proposed in the next section, which take the time-series aspect into consideration.

Multilayer perceptrons

Multilayer perceptrons (MLP) are a simple kind of artificial neural network, which consists of several layers, each counting one or more neurons. The neurons of a layer are connected with all the neurons of the previous one and no cycle are permitted, hence being classed as a *feedforward* network. When used as a function approximation, the first layer (input) contains the parameters of the target function and the last (output layer) represents its result for a given set of inputs. Between them, one or more hidden layers are used to model the behavior of the target function. Figure 5.2 illustrates an MLP used as a surrogate of a decision module objective in our problem.

The function computed by each neuron is basically a weighted sum of the application of its *activation function* applied to the value of each neuron of the previous layer. By considering the neuron j on layer l, with activation function g, its value is computed as in eq. (5.1). Here we denote N_{l-1} the number of neurons in the previous layer, $w_{l,i,j}$ the weighted accorded by this neuron to the neuron i and the last weight, $w^0_{l,j}$ representing the *bias* added to the final value. As the same method is used for each neurons of both hidden and output layers, the final value of a output neuron is simply given by the same equation and use the values of the last hidden layer. A small difference is the activation function used for the output neurons, which is often different than g and in practice is usually the identity function in this kind of usage.

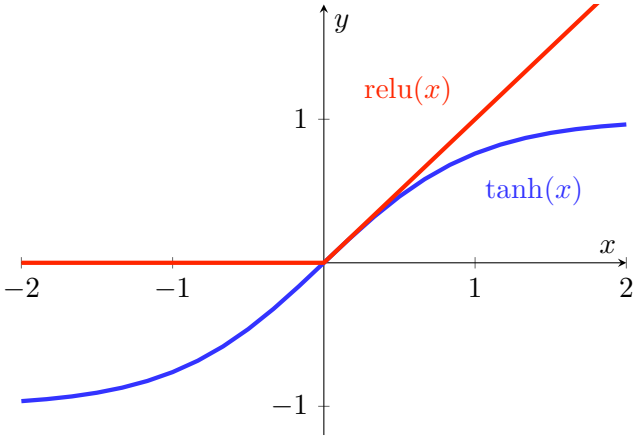


Figure 5.3 – Representation of *tanh* and *relu* activation functions for multilayer perceptrons.

$$v_{l,j} = w^0_{l,j} + \sum_i^{N_{l-1}} g(v_{l-1,i}) \cdot w_{l,j,i} \tag{5.1}$$

We do not aim to enter too much in the details of the implementation of MLP here, but instead highlight the most important aspects for using it. As other artificial neural networks, they are used in two distinct phases. First, a training phase is required to determine the values of the weights and biases. For this purpose, different solvers are used in the literature, but all have the same goal: based on a training set (combination of inputs and expected outputs), they progressively tune the weights to minimize the prediction error on this set.

Artificial neural networks in general are notoriously known for the many model parameters, called hyperparameters, to carefully select in order to obtain good results [168]. In addition to the choice of a solver of its internal parameters (learning rate, number of iterations and many others), the activation function must be chosen carefully as well as the shape of the hidden layers (number of layers and neurons in each). Because of the way a solver manipulates the neural network during the learning phase, not all functions are usable for playing the role of activation function. The rectified linear unit (*relu*) and the hyperbolic tangent (*tanh*) are among the most popular nowadays, showing good results in many kind of problems. Despite its convoluted name, *relu* is simply the minimum between 0 and the input, as represented in fig. 5.3.

Radial-basis functions

A radial-basis function by itself is a function which only depends on the *distance* between a vector and an origin. They are usually denoted φ, with the latter properties satisfied if $\varphi(x) = \varphi(||x||)$. Radial-basis functions are commonly used to approximate arbitrary functions, through several methods denoted generically RBF. The central idea is to use a set of origins together with a weight associate to each. The equation solved by the RBF is given in eq. (5.2), in which x is the input vector, x_i and w_i respectively the coordinate and the weight of each the N origins used. Similarly to MLP, a global bias w_0 may optionally be used.

$$f'(x) = w_0 + \sum_{i}^{N} \varphi(||x - x_i||) \cdot w_i \tag{5.2}$$

This usage was initially proposed as a kind of artificial neural network, which explains the similarity with an MLP having a single hidden layer of N neurons. Contrary to MLP, this number of neurons (or origins, depending the view used) can be chosen to match the number of known solutions if the set is relatively small (a few thousands or less) [172]. If this set is too large, a fixed number is used instead for N to keep training and approximation fast. The model is simpler when each known solution correspond to a single origin, as the training only consists in finding the best weights w_i for each solution input x_i. RBF of reasonable sizes are easier and cheaper to train than the general case of MLP. In addition, they are controlled by less hyperparameters, with the choice of the radial-basis function itself being the only significant one when using as much origins as training data.

Similarly to MLP, several functions are commonly used, such as cubic splines, quadratics or Gaussian ones, and some of them are controlled by an additional hyperparameter. While Gaussian function is often used, we focus here only on the cubic function ($\varphi(x) = ||x||^3$). The cubic function lead to good results in many surrogate-based optimization works, including high dimensional cases (≥ 100 solution space dimensions) and does not requires to tune an additional hyperparameter contrary to a Gaussian function [160].

5.3 Approach description

The approach presented in this chapter consists in two parts: a MOEA adapted for using surrogate models with low evaluation budget and the surrogate model itself. First, we propose a novel method, adapted to low evaluation budgets and based on Strength Pareto Evolutionary Algorithm 2 (SPEA2) [207], a well known multi-objective genetic algorithm. Then, two surrogate models, adapted to online usage and the high-dimensional nature of the time series are presented

5.3.1 Adapting SPEA2 for surrogate objectives functions

SPEA2 [207] is a genetic algorithm, designed to approximate Pareto front of multi-objective black-box problems. Its individual selection function is designed to favor both the improvement of the solutions towards the Pareto front and the diversity of the solutions to cover as much of the front as possible.

Similarly to Karakasis and Giannakoglou [98], a few additions are proposed to allow the MOEA to handle a mix of individuals with either evaluated (by the way of the costly objective functions) or approximated objective vectors. Each time a costly evaluation of an individual is performed, it is added to a set of known data points $\mathcal{P}$ used by the surrogate models. Contrary to existing approaches however, the surrogate methods are not considered to be reliable, in the sense that they may fail to give an estimation of the objective value. The causes of a failure, called thereafter a *miss*, are fully controlled by the surrogate method but may consist for instance in a lack of local data near the estimated solution.

To take into account the possibility of an approximation miss, surrogate function f' objective space is augmented by a special value $f' : \mathcal{S} \to \mathbb{R} \cup \{Miss\}$. Handling surrogate model misses is expected to be beneficial to the overall optimization, by avoiding to take into account approximations which are known to be of low quality. In such situation,

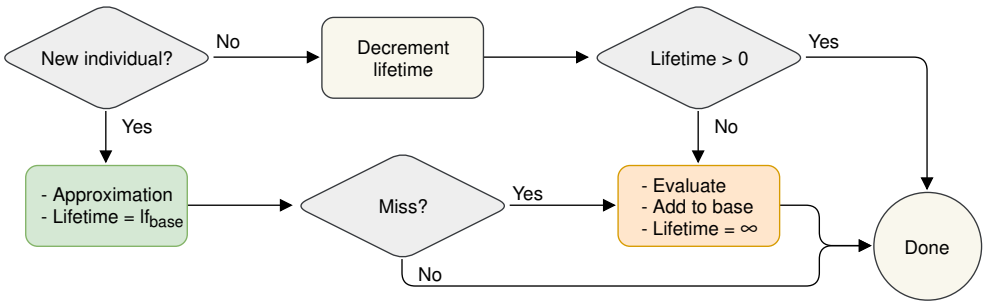


Figure 5.4 – Flowchart of the proposed algorithm, replacing the usual individual objectives evaluation in SPEA2 to handle approximation method.

it may be better to evaluate the solution immediately, which will additionally feed the knowledge base of surrogate model.

Two sets of modifications are proposed on the original SPEA2 algorithm. The first, presented as the *naive integration*, only consists in minimal changes to allow approximation of objectives of individuals and their evaluation with costly objective functions when some criterion is met. Then a set of improvements are detailed to handle mix of evaluated and approximated individuals, forming a new algorithm named USPEA2.

Naive integration

A simplified version of the SPEA2 algorithm with a few modifications is given in algorithm 8. The fitness assignment function and selection operator (*setSpea2Fitness()* and *spea2Selection()* in the listing) are not detailed here, but are the same as described in the original SPEA2 article [207]. The initialization of the population, denoted by the *initialPopulation*() function in the algorithms, is detailed later in section 5.4.2. Modifications to the original algorithm are highlighted in blue. They provide an integration of the utility approximation functions in SPEA2. The flowchart in fig. 5.4 gives an overview of the approximation or evaluation process replacing the usual individual objectives evaluation, corresponding roughly to the detailed algorithm 7. The utility values of each individual are approximated if possible, otherwise directly evaluated by interrogating each decision module.

When an individual is approximated using the surrogate models, a *lifetime* is associated to it with an initial value denoted lf_{base}. At each generation, the lifetime of all individuals is decremented. When it reaches 0, a complete evaluation of the individual objectives is performed. If any of the surrogate models fails to approximate the utility value (miss), the individual is immediately evaluated. Each time an individual is evaluated with the costly objective functions, it is also added along with its real utility to the knowledge base $\mathcal{P}$ of each surrogate. The number of generations is not the stopping criterion here, contrary to many genetic algorithms. Instead, the number of costly evaluations (sum of both decision modules) performed from the beginning is used. Depending on the quality of the surrogate and of the offspring scheme, the algorithm could take an unknown amount of generations to reach the maximum number of costly evaluations. Hence, to ensure the time spent by the algorithm is negligible compared to the evaluation functions, a mechanism forces at least one individual to be evaluated at each generation, as shown in algorithm 8. If no evaluation has been performed by the *evaluateRequired()* function, the best approximated individual of the population according to the SPEA2 fitness value is directly evaluated.

The detailed offspring generation algorithm is given in algorithm 6 for completeness.

```
Function generateOffspring(population)
    offspring ← ∅ ;
    for i ∈ {1, ..., populationSize} do
        ind1 ← binTournament(population) ;
        ind2 ← binTournament(population) ;
        ind1.valid ← false ;
        ind2.valid ← false ;
        if random(0, 1) ≤ pb_cross then
            offspring.appendAll(
              onePointCx(ind1, ind2)) ;
        else
            for ind ∈ {ind1, ind2} do
                mut = uniformMutate(ind,
                  pb_indmut);
                offspring.append(mut) ;
            end
        end
    end
    return offspring ;
```

Algorithm 6: Generation of the offspring population.

```
Function evaluateRequired(population,
  archive)
    foreach ind ∈ population ∪ archive do
        ind.lifetime ← ind.lifetime - 1 ;
        if not ind.valid then
            approximate(ind) ;
        end
        if ind.lifetime = 0 then
            u_pdm ← f_pdm(ind.value) ;
            u_itdm ← f_itdm(ind.value) ;
            ind.utility = (u_pdm, u_itdm) ;
            ind.lifetime ← ∞ ;
        end
        ind.valid ← true ;
    end
Function approximate(ind)
    u_pdm ← f'_pdm(ind.value) ;
    u_itdm ← f'_itdm(ind.value) ;
    ind.utility ← (u_pdm, u_itdm) ;
    if u_pdm = Miss ∨ u_itdm = Miss then
        ind.lifetime ← 0 ;
    else
        ind.lifetime ← lf_base ;
    end
```

Algorithm 7: Addition to SPEA2 for mixed approximation and evaluations.

Any pairs of individuals, selected using binary tournament with replacement, are either mated (with a probability pb_{cross}) or both mutated (probability $1 - pb_{cross}$). The mating function used here is a one-point crossover. A mutation consists of an independent probability pb_{indmut} for each time step of the individual to be replaced by a random value from a uniform distribution in the range of acceptable values $\mathcal{W}$.

USPEA2

SPEA2 gives good results for a wide range of problems with two or more objectives. However, it was designed to use reliable utility functions. In our approach, the utilities of an individual are either evaluated directly by the decision modules or by a surrogate function. In the first case, the utility is reliable as two evaluations of the same profile are guaranteed to return the same values during the optimization. In the second case however, the approximated utility is not reliable, as it may change if the individual is evaluated. Such approximated utility is therefore considered *unreliable*, as well as the individual to which the approximation is associated.

Unreliable utilities may lead SPEA2 to perform badly in some scenarios. An individual with approximated utility values may dominate an evaluated, reliable individual. In this situation, the unreliable individual is likely to replace the evaluated one. If the estimated individual is finally evaluated with lower utilities than the eliminated one, this results in a decrease of the overall individual pool quality and potentially in a deterioration of the Pareto front approximation.

To overcome this problem, we propose a variant of the SPEA2 algorithm, named USPEA2, for Unreliability-SPEA2. Figure 5.5 illustrates the algorithm, which is detailed in algorithm 9, with modifications compared to SPEA2 highlighted. It consists in the introduction of a second archive of individuals, called *reliable archive*, containing only

```
Function optimizeSpea2()
    archive ← ∅ ;
    population ← initialPopulation(populationSize) ;
    ended ← false ;
    while not ended do
        ended ← endingCondition() ;
        if ended then
            population ← {ind|ind ∈ population, ind.lifetime = ∞} ;
            archive ← {ind|ind ∈ archive, ind.lifetime = ∞} ;
        else
            evaluateRequired(population, archive) ;
            if no evaluation performed then
                evaluateBestIndividual() ;
            end
        end
        setSpea2Fitness(population, archive) ;
        archive ← spea2Selection(population, archive, archiveSize) ;
        if not ended then
            population ← generateOffspring(archive) ;
        end
    end
    return archive ;
```

Algorithm 8: Simplified algorithm of SPEA2. Lines in blue are additions to provide a naive support of the utility approximation functions.

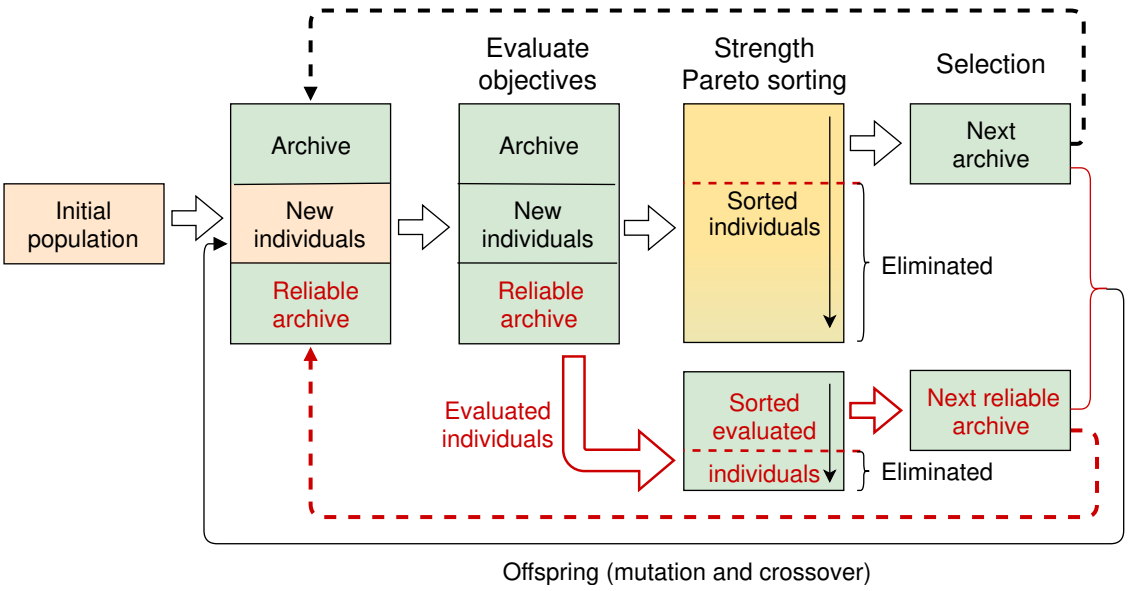


Figure 5.5 – Schematic view of USPEA2 algorithm with modifications compared to SPEA2 highlighted in red.

```
Function optimizeUspea2()
    archive ← ∅ ;
    reliableArchive ← ∅ ;
    population ← initialPopulation(populationSize) ;
    ended ← false ;
    while not ended do
        ended ← endingCondition() ;
        if not ended then
            evaluateRequired(population, archive) ;
            if no evaluation performed then
                evaluateBestIndividual() ;
            end
        end
        uniqueArchive ← unique(archive ⊎ reliableArchive) ;
        setSpea2Fitness(population, uniqueArchive) ;
        archive ← spea2Selection(population, uniqueArchive, archiveSize) ;
        reliablePopulation ← {ind|ind ∈ population, ind.lifetime = ∞} ;
        uniqueReliableArchive ← {ind|ind ∈ uniqueArchive, ind.lifetime = ∞} ;
        setSpea2Fitness(reliablePopulation, uniqueReliableArchive) ;
        reliableArchive ← spea2Selection(reliablePopulation, uniqueReliableArchive,
          archiveSize) ;
        if not ended then
            population ← generateOffspring(archive) ;
        end
    end
    return reliableArchive ;
```

Algorithm 9: Simplified algorithm of the Unreliability-SPEA2 (USPEA2) approach. Changes from SPEA2 are highlighted in blue.

evaluated solutions. Individuals from this second archive allow to maintain constantly a set of good and reliable solutions even if some of them are temporarily dominated by unreliable individuals. At each generation, after the fitness assignment and selection of the next archive in accordance with the classic SPEA2 scheme, an additional selection is done only on evaluated solutions, using the same selection operator. To avoid the introduction of a large number of duplicated individuals, which is not addressed efficiently by SPEA2, uniqueness of individuals is ensured when the two multi-set archives are merged.

5.3.2 Surrogate models for time series

Evaluating the quality of a potential solution $p \in \mathcal{S}$ is achieved by the way of the black-box objective functions $f_{itdm}(p)$ and $f_{pdm}(p)$. To approximate them, a surrogate model has the data base of previously evaluated solutions $\mathcal{P}$, with their power values normalized for practical reasons ($\mathcal{W} \rightarrow [0, 1]$).

Giving its definition, a power profile in $\mathcal{S}$ appears as a N-dimension vector, each dimension being a time step and the value for this dimension $\in \mathcal{W}$ being the power at this time step. Classic surrogate methods presented earlier, such as RBF and Multilayer Perceptron (MLP), can be used as they are known to work with high-dimensional solution space. However, they are not designed to leverage the *time series* aspect of the power profiles.

To the best of our knowledge, no existing work already focused on optimizing costly function defined on time series space. Despite the many areas where time series are studied, their use cases are usually quite different, such as the prediction of time series evolution. Therefore, we propose new approaches to perform utility approximation adapted for long profiles, inspired by works on signal and time series processing. These methods, contrary

to existing surrogate methods, may fail to return an approximation for some inputs (*miss*). As argued previously, we believe that forcing a costly evaluation because the surrogate method considers lacking local data to be accurate is not necessarily bad. Quite the contrary, it helps improving the quality of the surrogate around this new solution and prevent to return very inaccurate values which could impact negatively the overall optimization.

Average-distance method

By assuming that small differences between power profiles generally imply small differences in their utility values, a possible way to achieve utility approximation is by studying *similarity* between profiles. A simple way to determinate similarity between vectors is to use a well-known distance function.

The proposed method, named simply Average Distance (or AD) is similar in some aspects to a weighted k-nearest neighbors method [58]. It consists, first, in selecting profiles among the known and already evaluated ones which are closer than a given distance r_{close} from the unknown profile to estimate. The approximation is then performed by doing a weighted average of utility values of the close solutions, with a weight depending of their relative closeness. Based on an initial study, it appeared that euclidean distance was not always appropriate to compare profiles. In order to penalize some important local differences (*e.g.* a peak in one of the power profiles), a different measurement is used, based on the average of the square differences at each time step. As this is equivalent to mean square error (MSE) defined in statistics, it is used here and qualified by abuse of language as the distance between two profiles.

The complete algorithm is detailed in algorithm 10. The approximation of an individual p is considered as a miss when less than n_{close} are known in a radius of r_{close} around p.

Function *distanceUtilityApproximation(target,* r_{close}*,* n_{close}*,* $\mathcal{P}$*,* lf_{base}*)*
  $\mathcal{P}_{close} \leftarrow \{p | p \in \mathcal{P}, \text{mse}(p, target) < r_{\text{close}}\}$;
  if $|\mathcal{P}| \geq n_{close}$ **then**
    approx $\leftarrow distanceWeighted(target, r_{\text{close}}, \mathcal{P}_{close}, \text{mse})$;
    return *approx* ;
  else
    return *Miss* ;
  end
Function *distanceWeighted(target, radius,* $\mathcal{P}_{close}$*, distanceFunc)*
  totalWeight $\leftarrow \sum_{p \in \mathcal{P}_{close}} radius - distanceFunc(p, target)$;
  return $\sum_{p \in \mathcal{P}_{close}} objective(p) \frac{radius - distanceFunc(p, target)}{totalWeight}$;

Algorithm 10: Average-distance (AD) approximation algorithm, for a given *target* profile and using the known solutions $\mathcal{P}$ with their utility values given by *objective*.

Multiresolution Haar transform distance method

A power profile may be considered, instead of an ordinary N-dimensional vector, as a time series of power values. Time series are well studied in various areas, such as in economics and meteorology. Particularly, a lot of works exist which deal either with finding similarities between several time series [2] or predicting next values of time series using past data [18].

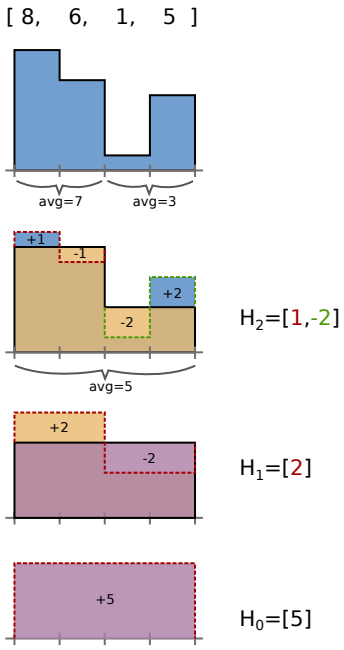


Figure 5.6 – Illustration of the transformation of a time series $X = (8, 6, 1, 5)$ using Haar wavelets, resulting in flatten representation $HC_2 = (5, 2, 1, -2)$.

While there is no existing work usable directly for building surrogate models of function defined on time series, similarity detection of time series shares some challenges with our problem. Several works already demonstrated some qualities of wavelet transform for similarity detection and similarity indexing of time series and more specifically of Haar wavelets [2, 29, 30]. Wavelet transforms provide an elegant way to achieve dimensionalty reduction in time series, by resulting in a multiresolution representation.

We first need to introduce some notations and properties of the Haar wavelet transform, as well as a definition of a partial distance between Haar representations. Then, the surrogate method itself is described.

Overview of the Haar transform Transforming a series of $N = 2^k$ values using Haar wavelet gives a new series of same size, containing the coefficients of the wavelet for different combinations of scale and shift. These coefficients are commonly grouped by *sequence* [177], each containing the coefficients for a given scale (or frequency). Figure 5.6 shows the application of Haar wavelet transform, with a few simplifications on the actual resulting values for illustration purpose.

A total of $k + 1$ groups are obtained this way, denoted by $H_0, \ldots, H_k$. The first, H_0, is a single value proportional to the global average of the series. The other groups $H_i, i \in \{1, k, \ldots,\}$ have a size given by $|H_i| = 2^{i-1}$. Each sequence H_i contains information on the original series with a temporal resolution doubled compared to H_{i-1}.

Let X, Y be two series of size $N = 2^k$. The concatenation of the sequence $H_0, \ldots, H_i$ of the Haar transform of X, called here its *flatten notation*, is denoted by $HC_i(X)$.

Haar transform has several interesting properties for similarity finding:

- the Euclidean distance between two profiles X, Y is the same than the euclidean distance between their flatten Haar representation $HC_i(X), HC_i(Y)$ [30]
- each depth of the tree representation is tied only to local features of the signal
- the first components of the Haar representation depend only on the *low frequency*

features, while each successive depth increase the high frequency details, allowing efficient dimensionalty reduction
- Haar transform is fast to compute, having a time complexity in $O(n)$ [30], with n the length of the signal

Haar-based distance function The Haar transform of signals may be used to define a new partial distance function. As mentioned previously, the Euclidean distance between two signals is the same as the Euclidean distance between the flatten representation of the Haar transform of those signals. With *dist* being the Euclidean distance function, and $X, Y \in \mathbb{R}^l$ two signals of length $l = 2^k$, we have:

$$dist(X, Y) = dist(HC_k(X), HC_k(Y))$$

We introduce a *Haar distance* function, denoted $hdist_d$, which give the distance between the first d sequences of two Haar transform:

$$hdist_d(X, Y) = dist(HC_i(X), HC_i(Y))$$

The Euclidean distance is monotonically increasing when new elements are added to each vector, such as:

$$dist(u_{0\cdots l-1}, v_{0\cdots l-1}) \leq dist(u_{0\cdots l}, v_{0\cdots l})$$

Two properties must be highlighted here. For two signals $X, Y \in \mathbb{R}^l$, with $l = 2^k$:

1. with $d = k$, the maximum depth, the Euclidean distance and the Haar distance are equal (eq. (5.3))
2. Haar distance is monotonically increasing with the value of d (eq. (5.4))

$$dist(X, Y) = hdist_k(X, Y)) \tag{5.3}$$

$$\forall i, j \in \{0, \ldots, 2^k\}, i \leq j, hdist_i(X, Y) \leq hdist_j(p, q) \leq dist(p, q) \tag{5.4}$$

These properties are exploited in a new surrogate model, leveraging both dimensional reduction by using Haar distance and local precision around known solutions by using the monotonic convergence towards the real Euclidean distance.

Haar-based surrogate We propose here a new surrogate method, named Multiresolution Haar Transform (MHT). It uses the same basic idea of Average Distance method, but exploit the Haar transform of the solutions at multiple temporal *resolutions*. The full algorithm is detailed in algorithm 11. Basically, it looks for the profiles closer than a fixed radius r_{close} to the one to approximate according to the Haar distance at a given depth. This depth, starting from 0, is progressively increased, hence reducing the number of profiles close enough to be considered. It stops when the increase of depth results in a close solutions set $\mathcal{P}_{close}$ smaller than a given threshold n_{close}. The approximation is then calculated based on the average of the objective values of the close solution set, weighted by their closeness. As the Haar wavelets are only defined on series of length being a power of 2, the power profiles are filled with 0 up to the nearest valid length.

To be easier to understand, the full algorithm given in algorithm 11 is simplified and not optimized. Some optimizations may be used to achieve low average complexity. Mainly, profiles (known solutions and a target profile) may be stored along with their Haar

Function *haarUtilityApproximation(target,* r_{close}*,* n_{close}*,* $\mathcal{P}$*,* lf_{base}*)*
  depth $\leftarrow 0$;
  $\mathcal{P}_{close} \leftarrow \mathcal{P}$;
  $\mathcal{P}_{accepted} \leftarrow \emptyset$;
  while $depth < k \wedge |\mathcal{P}_{close}| \geq n_{close}$ **do**
    $\mathcal{P}_{close} \leftarrow \{p | p \in \mathcal{P}_{close}, hdist_{depth}(p, target) < r_{\text{close}}\}$;
    if $|\mathcal{P}_{close}| \geq n_{close}$ **then**
      $\mathcal{P}_{accepted} \leftarrow \mathcal{P}_{close}$;
      depth $\leftarrow depth + 1$;
    end
  end
  if $\mathcal{P}_{accepted} \neq \emptyset$ **then**
    approx $\leftarrow distanceWeighted(target, r_{\text{close}}, \mathcal{P}_{accepted}, hdist_{depth})$;
    return *approx* ;
  else
    return *Miss* ;
  end

Algorithm 11: Haar-based utility approximation algorithm, approximating the objective value for *target* profile and using the known solutions $\mathcal{P}$. The function *distanceWeighted()* is defined in algorithm 10.

transform result, avoiding re-calculation at each distance measurement. This makes Haar distance computing complexity $O(n)$, with n the profiles length.

Therefore, the complexity of Haar surrogate algorithm is dominated by its main loop having a maximum of $\log_2(n)$ iterations. Each iteration requires to evaluate the Haar distance with a subset of the known profiles set, for a worst-case time in $O(n \cdot m)$, m being the number of known solutions. The overall, worst case time complexity is thus $O(n \log(n) \cdot m)$.

However, we can expect better performance in average. The set of the considered solution at iteration i of the main loop is always a subset of the close solutions found at iteration $i-1$. In addition, the depth of the Haar distance is increasing at each generation, starting from constant-time complexity (distance between two 1-dimension vectors) and doubling at each iteration. Thus, the most costly distance computing occurs with the smaller set of retained solutions.

5.4 Evaluation methodology

To evaluate our approach, a practical optimization problem must be targeted. While some multi-objective functions are commonly used in the MOEA field for evaluation purpose, such as ZDT [206] or DTLZ [47], they do not capture some characteristics of the problem exposed here. Particularly, they are not defined on time series, which is our primary target.

Our approach is designed to use black-box objective functions for both IT and electrical decision modules. Hence its evaluation requires to specify the models and algorithms used for the concrete modules. To evaluate the performances of a purely multi-objective optimization method in a complete and extensive way, knowing the real Pareto front of the problem is ideal. It represents the best achievable set of trade-offs and therefore a solid and unquestionable point of comparison.

We target a medium scale datacenter powered partially by on-site RES and batteries. A scheduling algorithm from the literature, such as RECO [27], is usable as the IT decision

module. The RECO algorithm consists in a batch task scheduling algorithm under power envelope constraint, targeting heterogeneous datacenter and leveraging Dynamic Voltage and Frequency Scaling (DVFS) to modulate the power consumption. Finding the real Pareto front of the overall problem with such detailed model, however, is hardly feasible.

Hence a simplified model of datacenter and electrical infrastructure is proposed first. It is designed to allow a Mixed-Integer Linear Programming (MILP) formulation of the overall optimization problem with reasonable solving time, providing a very accurate Pareto-optimal set of solutions, while keeping most characteristics as expected in a similar real-world problem. This model is therefore used to perform extensive evaluations of the proposed heuristics. To validate the main findings, additional experiments are performed by using the RECO scheduling algorithm in place of the simplified IT model.

After a detailed presentation of this model, its various parameters are discussed in order to propose realistic instances for evaluation purpose. This finally leads to the definition of scenarios, consisting of variations of environmental parameters (weather forecast and battery state of charge).

5.4.1 Simplified model for evaluation

The practical optimization problem proposed for the evaluation is composed of two sub-models, tied together only by the way of provided or consumed power denoted p_t for each time $t \in \{0, \dots, T-1\}$. On the one hand, the electrical model includes the RES, ESD (such as batteries) as well as a connection to the electrical grid through an electricity provider. On the other hand, the IT model describes the tasks, the machines and their power consumption. Both models are linked by the resulting power availability from the electrical part being equal to the power consumed by the IT part. An instance of the model concerns a single time window of T time steps, of fixed duration δ each. A summary of the symbols used in the model is given in table 5.1.

Electrical model We consider a set of RES, producing together, at each time step $t \in \{0, \dots, T\}$, an average power of r_t. In addition, batteries may be used to store some energy. These batteries have a total capacity bat_{max} energy units, with an initial charge of bat_{init} at the beginning of the window, as given by eq. (5.5a). The remaining charge at the beginning of a time step t is denoted by c_t and is constrained by eqs. (5.5b) and (5.5c). The whole electrical system is connected to the electrical grid. The power bought from or sold to the grid to it at a time t is given by g_t. The sign of g_t indicates if the power is bought ($g_t > 0$) or sold ($g_t < 0$). Finally, the maximum power made available by the electrical part is the sum of the renewable production, battery charge or discharge and grid usage, as given by eq. (5.5d).

$$\forall t \in [0, T), \begin{cases} c_0 = bat_{init} & \text{(5.5a)} \\ c_t \geq 0 & \text{(5.5b)} \\ c_t \leq bat_{max} & \text{(5.5c)} \\ p_t \leq r_t + (c_{t+1} - c_t)\delta + g_t & \text{(5.5d)} \end{cases}$$

The utility of electrical part represents the environmental impact of a decision, calculated through several criteria. A first part, in eq. (5.6), depends on the grid usage, with each unit of energy bought decreasing the utility by ghg_{buy} and each unit sold increasing

it by ghg_{sell}.

$$u_{grid} = \sum_{t=0}^{T-1} \begin{cases} -ghg_{buy} \cdot \delta \cdot g_t & \text{if } g_t \geq 0 \\ -ghg_{sell} \cdot \delta \cdot g_t & \text{otherwise} \end{cases} \tag{5.6}$$

As a battery life highly depends on the amount and the depth of charge and discharge cycle[33, 59, 186], a second part of the utility reflects this additional aging cost, as in eq. (5.7).

$$u_{aging} = cost_{aging} \sum_{t=0}^{T-1} |c_{i+1} - c_i| \tag{5.7}$$

Finally, a third part represents a *potential saving* coming from the energy stored in the battery at the end of the time window presented in eq. (5.8). The potential gain is calculated as if it saved the same amount of energy from being bought later on the grid, while taking into consideration the aging of batteries caused when the energy will be used.

$$u_{storage} = c_T(ghg_{buy} - cost_{aging}) \tag{5.8}$$

The total utility, for the electrical part, is therefore given by eq. (5.9) below.

$$u_{elec} = u_{storage} - u_{aging} - u_{grid} \tag{5.9}$$

IT model The IT part is modeled as a set of M identical machines, each consuming $power_{machine}$ when being used. The workload is considered as a fluid mass W, expressed in equivalent run time for a single machine (machine·time unit). This total mass of the workload W can be divided into an arbitrary number of chunks to be run on any machine at any time (eq. (5.10a)). It can be seen as a *divisible load* with no communication cost and a linear cost on the load. However, each part of the workload requires an entire machine during a full time step.

The number of machines used at a given time t is equal to the amount of tasks executed, noted w_t and constrained by eqs. (5.10b) and (5.10c). The minimum power required at every time step is therefore given by eq. (5.10d).

$$\forall t \in [0, T), \begin{cases} \sum_{t=0}^{T-1} \delta \cdot w_t \leq W & \text{(5.10a)} \\ w_t \geq 0 & \text{(5.10b)} \\ w_t \leq M & \text{(5.10c)} \\ p_t \geq power_{machine} \cdot w_t & \text{(5.10d)} \end{cases}$$

The utility of the IT part depends on the executed workload. We consider that the benefit for executing any workload is homogeneous through $gain_{task}$:

$$u_{base} = gain_{task} \sum_{t=0}^{T-1} \delta \cdot w_t \tag{5.11}$$

A second part of the utility is given by an additional gain when a task is submitted early. This gain, for each task, is at most $gain_{early}$ if the task is executed in the first slot, as given in eq. (5.12).

$$u_{early} = gain_{early} \sum_{t=0}^{T-1} \frac{(T-1-t)w_t}{(T-1)\delta} \tag{5.12}$$

Maximizing u_{early} is then equivalent to minimizing the flow time of the workload.

Finally, eq. (5.13) gives the overall IT utility.

$$u_{it} = u_{early} + u_{base} \tag{5.13}$$

Decision modules and centralized Pareto approximation

The IT and electrical models can be used to derive a centralized MILP formulation. However the optimization problem has two different objectives: maximization of both u_{itdm} and u_{pdm}. To obtain an approximation of the Pareto front of solutions, two steps are required. In a first phase, the extreme solutions are obtained by optimizing each objective without any constraint on the other. It results in two pairs of utility values, $(u_{pdm}^{min}, u_{itdm}^{max})$ when ITDM utility is optimized and $(u_{pdm}^{max}, u_{itdm}^{min})$ when it is PDM utility. Then, a desired number of solutions distributed along the Pareto front are obtained by maximizing one of the utility subject to an additional constraint on the second:

$$\begin{aligned} &\text{maximize } u_{itdm} \\ &\text{subject to } u_{pdm} \geq u_{pdm}^{min} + \alpha \left(u_{pdm}^{max} - u_{pdm}^{min} \right) \end{aligned}$$

By repeating this for different values of $\alpha \in [0, 1]$, a good approximation of the real Pareto front is provided with as many solutions as required.

Each infrastructure model can also be used to implement each decision module. By accounting only the constraints of the electrical infrastructure (respectively IT infrastructure), the utility of a power profile P is given by maximizing u_{pdm} (respectively u_{itdm}) with additional constraints:

$$\forall t \in [0, T), p_t = P_t$$

Complex IT model

The RECO algorithm [27] uses a finer and more realistic IT model. The computing infrastructure handles heterogeneous machines, with DVFS and power-on/power-off models taking into account the time and power required for booting up or shutting down the machines. It focuses on a workload of batch tasks with due dates, each having different requirements in terms of CPU.

5.4.2 Parameters for evaluation

The evaluation of our approach is done using several instances of the electrical and computing infrastructure models. Some parameters, such as the capacity of the battery bat_{max} or the duration of a time step δ, are given by an expression of higher level parameters. This helps making simpler the definition of the different *scenarios*. The values of the parameters related to GHG emission and renewable power production use existing studies and publicly available data to keep realistic figures. Table 5.1 provides a short summary of the values used for the symbols corresponding to model parameters.

Time window and time step For all the experiments, a fixed time window of 72 hours is used, starting at midnight. The number of time steps T, however, is set to different values across the experiments, being either 20, 80 or 320. The time step duration δ thus depends on T such as $\delta = 72/T$ hours.

Symbol	Meaning	Value in final evaluation
ghg_{buy}	Cost (in greenhouse gas (GHG) emission) of an energy unit from the grid	0.3 kg CO_2-eq /kWh
ghg_{sell}	Resell price of an energy unit to the grid provider	0.15 kg CO_2-eq /kWh †
$cost_{aging}$	Cost due to battery aging when battery is charged or discharged.	0.0483 kg CO_2-eq /kWh
$gain_{task}$	Benefit when a unit of workload is executed in the time window.	1
$gain_{early}$	Maximal additional benefit when a unit of workload is executed early	0.2
T	Number of time steps in a time window	{20, 80 or 320}
δ	Duration of a single time step	$72/T$
r_t	Renewable power available at time t	-
c_t	Energy in the batteries at beginning of time t	-
bat_{max}	Total capacity of the batteries	*
bat_{init}	Energy in the batteries at $t = 0$	$bat_{max} \cdot bat_{soc}$*
g_t	Power bought from ($g_t > 0$) or sold to ($g_t < 0$) the grid at time t	-
$power_{machine}$	Machine consumption when used	200 W
W	Total amount of workload to execute in the time window	*
M	Number of available machines	100
w_t	Amount of workload executed at time t	-
p_t	Power used at time t	-

Table 5.1 – Summary of symbols used in the electrical and IT models.
† Except for the specific experiments showing the impact of this value in approach results.
* Calculation detailed in section 5.4.2.

Workload and machines The amount of workload considered is based on the maximal work that may be executed by the datacenter if all the machines are used during the whole time window. A factor is then applied, to consider an average load of $\overline{Load} = 70\%$. Therefore, we can define its value using eq. (5.14) below.

$$W = \overline{Load} \cdot \delta T \cdot M \tag{5.14}$$

The experiments account for $M = 100$ machines. Each is consuming an average power of $power_{machine} = 200\,\text{W}$ under load, for a datacenter peak consumption of 20 kW. The ITDM utility account for a benefit for each unit of workload executed in the time window $gain_{task}$ and additional benefit which is higher when the task is executed near the beginning of the time window. These parameters are fixed to values in arbitrary units, $gain_{task} = 1$ and a maximal benefit when executed early $gain_{early} = 0.2$.

Battery capacity and initial state The maximum and initial capacity of the battery is chosen between several settings. The maximum capacity is set to provide enough energy

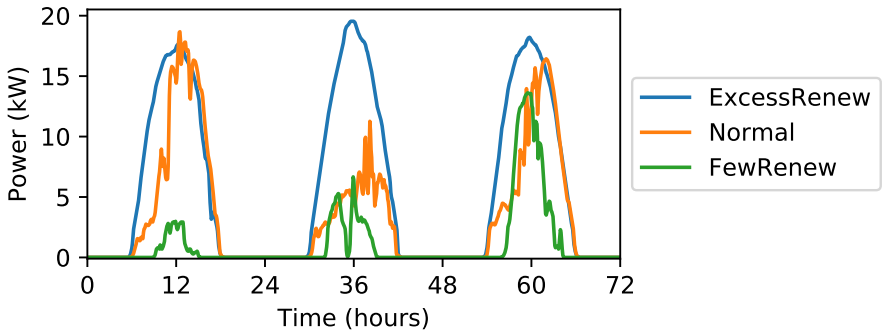


Figure 5.7 – Power production for the simulated photovoltaic (PV) power plant in California during the chosen periods.

to fully power all the machines during 12 hours. With the parameters chosen for the IT part, this gives approximately 240 kW h of batteries. This is a relatively high figure, but result in an initial investment of around $\$65,000$ based on the 2016 values [40]. This seems acceptable, staying in the order of magnitude of the budgets and operating cost of such a medium-scale datacenter.

$$bat_{max} = power_{machine} \cdot M \cdot 12 \tag{5.15}$$

The initial capacity is then simply set by using a coefficient bat_{soc}, such as $bat_{init} = bat_{max} \cdot bat_{soc}$.

Photovoltaic production data

We used the high quality synthetic photovoltaic production data from the U.S. National Renewable Energy Laboratory (NREL), contained in the Solar Power Data for Integration Studies dataset [144]. It contains, for hundreds of power plant locations, 5-minutes power data is available during all the year 2006.

We used the data from a power plant located at coordinates N34°85′ W120°45′, near the city of Santa Maria in California. This is a medium power plant, with a peak capacity of 4 MW (with a maximum of 3.93 MW reached in these data). For the need of our experiments, we choose only a few 3-days periods of interest inside the year-long data, with associated production shown in fig. 5.7:

- **ExcessRenew**: almost perfectly sunny days, starting 1st June 2006
- **Normal**: noisy days with the second one being very cloudy, starting 10th June 2006
- **FewRenew**: very cloud days during the winter, starting 1st January 2006

This power plant is scaled to fit the need of the considered data center. More precisely, we assume that the goal was to produce a yearly amount of energy equals to the yearly amount of energy consumed by the data center (considering the average load $\overline{Load} = 0.7$, as given above). The yearly average of power production is $\overline{PvProd} = 730.9\,\text{kW}$. Therefore, the scaling factor used on the PV data is $pvScaling = 1.915 \times 10^{-2}$, as given by eq. (5.16) below.

$$pvScaling = \frac{M \cdot \overline{Load}}{\overline{PvProd}} \tag{5.16}$$

Gas emission parameters

The electrical utility for a given profile is an amount of GHG emission avoided the given power during the time window of the profile. These values are given in equivalent kilogram of CO_2. It is rather difficult to obtain unquestionable values, as discussed in section 2.3, but realistic values from available sources are used. In this work, we do not focus on the total environmental impact of building such a green datacenter, considering it already built. Hence, even if their manufacturing is a pollution source [7, 151], the energy from photovoltaic panels is considered free of any environment impact.

Batteries We consider some Lithium Manganese Oxide (LMO) batteries, from the lithium-ion family. According to Hao et al. [82], the emission of GHG caused by their manufacturing is about 96.6 kg CO_2-eq /kWh. The environmental impact of using the batteries is modeled as described in section 2.3.3, considering a linear degradation of their lifetime with the charged or discharged energy. By using lifetime of $N^{cycle} = 1000$ full charge/discharge cycles, we found an amortized GHG emissions of $cost_{aging} = 0.0483$ kg CO_2-eq /kWh.

Electrical grid Accurate data of electrical grid GHG emissions are hard to find, especially with hourly or finer resolution. Several less-extensive datasets are made available by either an electrical grid operator or organizations (both governmental and nongovernmental). In the case of California, some operators like CISO give some information on the GHG emissions [25] every month. The California Environmental Protection Agency provides an annual GHG emission inventory [24] for multiple sectors, including electrical power generation. Based on the 2016 report, the average annual values for the year 2014, including energy imports, is 0.3 kg CO_2-eq /kWh. This value is used for ghg_{buy}.

Selling an excess of production on the electrical grid is expected to avoid some production of GHG. For one unit of renewable energy sold in the grid, the avoided GHG is likely less than the emission caused by the production of one unit on the overall grid. In addition to the various losses such as the conversion and transport ones, the grid is not always able to absorb excessive energy as the traditional power plants may already satisfy the demand at a given time. To account of this, the avoided GHG emission for selling energy to the electrical grid ghg_{sell} is considered here, on average, to be half of ghg_{buy}.

Complex ITDM settings

The experiments involving the RECO scheduler [27, 165] use the same parameters and workload than in the original paper. The computing infrastructure consists in 30 bi-processor machines of two kinds, with a workload composed of 1029 tasks of various duration and CPU requirements. They are submitted along a 48-hours time window, with a duration and submission using the synthetic workload generator described in section 3.2.

To use it as IT objective function in our approach, the power profile evaluated is used as the maximum power envelope and the utility (objective value) is computed as the percentage of tasks scheduled without violation of their due dates. In the experiments involving the complex IT model, we consider the same electrical model than for other experiments. However, to take into account the lower total consumption for the associated workload, the renewable production and battery capacity are scaled to 20% of the value used for the simplified IT infrastructure.

Name	PV production	bat_{soc}
ExcessRenew	Good	0.5
Normal	Bad	0.5
FewRenew	Worst	0.25

Table 5.2 – Definition of the scenarios used for the experiments.

Scenarios

The number of model parameter combinations is important, but the main goal of the experiments is to evaluate the performance of the approach in a few different conditions. We define therefore a set of 3 scenarios, consisting of variations of the photovoltaic panels production data and battery settings. Their names and the values used are given in table 5.2. They are chosen to have both some expected normal situation and edge cases where the problem space may be easier or more difficult to explore.

Meta-heuristic settings

The SPEA2 and USPEA2 algorithms, along with their mutation and crossover functions, are configured by several parameters. The values chosen for probability of crossover pb_{cross} and the independent probability of time step mutation pb_{indmut} are detailed in section 5.5.1, based on experimental results. Some other are not experimentally studied here.

The size of the archives and of the population, *archiveSize* and *populationSize*, are both set to 20. This relatively small value is chosen because a low number of costly objective function evaluation budget is aimed in this approach. A higher number of individuals in the population would require more evaluations, which is not desirable here. Lower values tend to limit too much the diversity of the individuals, which is required to have useful crossover.

As described in section 5.3.1, the mutation function used causes each time step of the concerned individual to be potentially replaced by a random value following a uniform distribution. The boundaries of the distribution are chosen to match the requirements of the decision modules while minimizing the explored solution space. By knowing the minimum and maximum power produced or consumed by each decision module (p_{min} and p_{max}), the range of the uniform distribution is given by:

$$[\max(p_{min}^{itdm}, p_{min}^{pdm}); \min(p_{max}^{itdm}, p_{max}^{pdm})]$$

In practice with the models used, the minimum is bounded to 0 (electrical infrastructure not providing power, computing infrastructure with all machines powered off). The maximum is unbounded for the PDM, but limited by the ITDM, when all machines are used simultaneously ($M \cdot power_{machine}$).

In the description of SPEA2 and USPEA2 algorithms, the individuals forming the population at the beginning of the first generation is abstracted by calling $initialPopulation(size)$ function. Two different initial population schemes are studied here.

The primary one is the *Random* scheme. It simply consists of a generation of *size* individuals for which every time step values is randomly selected using an uniform distribution. The bounds of the distribution are the same as for the mutation function.

The second scheme is called *Best*. While $size - 2$ individuals are generated similarly to the *Random* scheme, 2 individuals are provided by the decision modules themselves. Each module is supposed able to compute an optimal power planning for its own objective

(without any constraint related to the other's requirement). Therefore, the best power planning of each module is added to the initial population and is expected to be part of the real Pareto front as it maximizes one of the objectives.

To compare the results of different heuristics, the two multi-objective performance metrics detailed in section 2.1.1: hypervolume indicator and generational distance. The hypervolume indicator (abbreviated HV) gives the space covered between a set of non-dominated solutions and a pessimistic (Nadir) point, measuring both the diversity of the solution and the accuracy of the Pareto front approximation. For comparing the results of two approaches, the same Nadir point must be used. The same reference point is used for all the comparable results (same scenario and same number of time steps) and consists of the minimum values for each objective across all these results. For ease of interpretation, the normalized hypervolume (nHV) is used, considering the Pareto front obtained by the MILP for the same scenario and time window as a reference.

The generational distance (GD), also referred as *average distance to Pareto front*, measures only the accuracy aspect, but helps interpreting the hypervolume values. Similarly to hypervolume, the raw values of this metric are not easy to interpret as it depends on the scale of the each objective. Hence a normalization is performed on the objective values to scale each dimension of the objective space in $[0, 1]$ before computing the generational distance.

5.5 Preliminary experiments on approach parameters

Several parameters must be chosen for the optimization algorithms (SPEA2 and USPEA2) and the different surrogate methods.

- pb_{cross}, the probability of crossover/mutation
- pb_{indmut}, the probability for each time step value of a mutated individual to be modified
- lf_{base}, the initial lifetime of approximated objective values
- r_{close}, the distance considered as *close* for the Average Distance (AD) approximation and Multiresolution Haar Transform (MHT) approximation methods
- n_{close}, the minimum number of close known solutions to do an approximation (AD and MHT)
- number of neurons and structure of the hidden layers (MLP)
- activation function (MLP)

The behavior of the overall optimization process depends on the interaction between (U)SPEA2 and the surrogate models. For instance, the accuracy of the surrogate models affects the individual that will be preferred by the MOEA to survive, affecting which individuals will be eventually evaluated. In turn, the evaluated individuals affect the data base of known solutions, impacting the quality of the approximation performed by the surrogate. Therefore the precise impact of each parameter is expected to be tied to all the others. It would be rather impractical, especially for more complex problems, to evaluate the overall results of such approach for a large set of parameter combinations. Instead, we propose a simplified methodology to choose them.

The proposed methodology does not aim to find the optimal combination of their values for our precise evaluation scenarios (overtuning the parameters). To enforce that, only one of the evaluated scenarios is used for setting the parameters. In a first time, the two parameters of (U)SPEA2 (crossover probability and independent mutation probability) are studied, without considering any approximation method. Then for each surrogate approach, the impact of their parameters on the quality of approximation is studied, using

a prepared training set of power profiles and associated objective values (*i.e.*, independently from a MOEA). Finally, the initial lifetime of approximated individuals lf_{base} is studied by using all the previously chosen parameters, by using MOEA with surrogate method.

5.5.1 SPEA2 and USPEA2 parameters

First of all, the core parameters of the genetic algorithm need to be set to reasonable values, leading to good results without any objective approximation method. To evaluate the impact of their value on the overall results, SPEA2 is used with the scenario *Normal* and 80 time steps in the window (T). Their resulting metrics (hypervolume and average distance to Pareto front) are measured after 100, 1600 and 6400 evaluations. The probability of crossover pb_{cross} is chosen between 0 (mutation only) and 1 (crossover only). For the probability of individual gene mutation pb_{indmut}, it is chosen such as the average number of values (time step) mutated in an individual vary between 1 and T following a geometrical progression. As genetic algorithms results have an inherent part of randomness, especially for a low number of iterations, each combination of parameters is repeated 10 times. The values shown below are therefore the average metrics over the 10 runs.

Figure 5.8 presents all the results in the form of heatmaps. As expected from a genetic algorithm, results for a very low number of iterations are noisy and with low disparity, as shown in figs. 5.8a and 5.8b for 100 evaluations. With more iterations, some patterns appear.

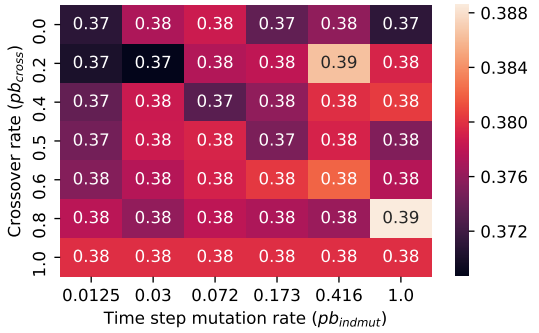
For the hypervolume indicator, worst results are reached when either pb_{cross} is close to 1.0 (no mutation) or when pb_{indmut} is close to 1.0 (mutation causing all the power profile to change), both for medium amount of evaluations (fig. 5.8c) and important amount (fig. 5.8e). The best results for 1600 evaluations are achieved with $0.03 \leq pb_{indmut} \leq 0.173$ and $0 \leq pb_{cross} \leq 0.6$. For a larger amount of evaluations, a lower value of pb_{indmut} seems preferable ($0 \leq pb_{indmut} \leq 0.072$) as well as a value of pb_{cross} closer to 0.5.

When it comes to the average distance to Pareto front (GD), the disparity is less important than hypervolume even for higher number of evaluations as presented in fig. 5.8d (1600 evaluations) and fig. 5.8f (6400 evaluations). A noticeable improvement is however observable with crossover rate pb_{cross} between 0.5 and 0.8 along with non-extreme values of pb_{indmut}.

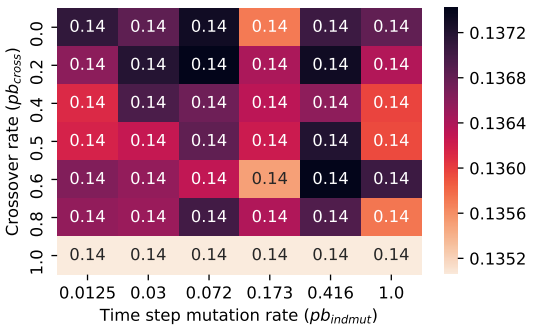
As our experiments will focus on low and moderated amount of evaluations, we chose values in favorable range for both hypervolume and GD at 1600 evaluations, while taking into account the results for 6400 evaluations in order to choose in this acceptable range. In all the following experiments, we use values of $pb_{cross} = 0.5$ and $pb_{indmut} = 0.072$.

5.5.2 Surrogate methods parameters

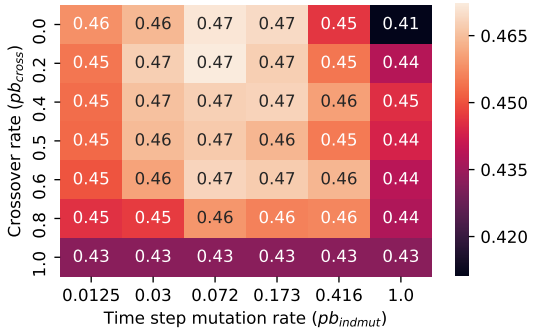
Four different surrogate methods are explored. The RBF with *cubic* function is parameter-free, but each of the three others has various parameters to tune for them to perform well. The use of surrogate models of a complex objective function is well studied for single-objective evolutionary optimization, especially when the surrogate model is constructed off-line. In such case, the surrogate (or approximation function) may be studied as a prediction function and a lot of common statistical analysis may be performed to evaluate the quality of the approximations. For instance, the R^2 or the root-mean-square deviation (RMSD) of the surrogate may be computed using a well distributed set of samples. It may be assumed that a surrogate model leading to higher R^2 (or lower RMSD) will perform better. In the case of MOEA and/or online methods, it is more difficult to evaluate an approximation method and very little literature exist on the topic.



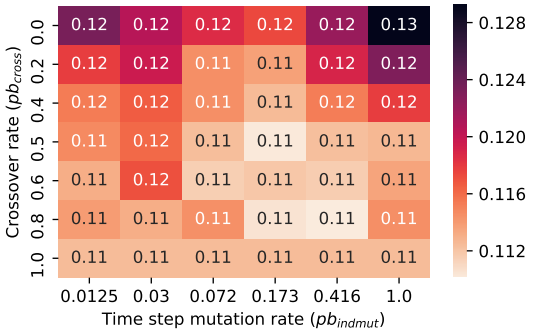
(a) Hypervolume for 100 evaluations



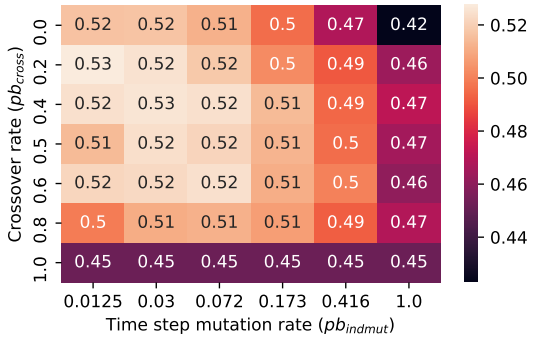
(b) Average distance to Pareto for 100 evaluations



(c) Hypervolume for 1600 evaluations



(d) Average distance to Pareto for 1600 evaluations



(e) Hypervolume for 6400 evaluations

(f) Average distance to Pareto (GD) for 6400 evaluations

Figure 5.8 – Performance of SPEA2 depending on crossover rate and independent mutation probability, for different evaluation budgets. Higher is better for hypervolume, lower is better for generational distance (GD).

In our context, evaluation of our surrogate functions is challenging for the following reasons.

- As mentioned previously, the overall performances of MOEA with surrogate methods likely depends on complex interactions between both parts which are not captured by looking only at the quality of the approximations
- Evaluating an approximation method independently requires a set of solutions (power profiles with their utility values) representative of the solutions that would be produced by the MOEA.
- The two new approximation methods proposed may either return an approximated value or indicate a "miss", considering that the known solutions set is not appropriate to perform the approximation. This leads to at least two different metrics for measuring the quality of an approximation method: the accuracy of the approximations themselves and its *miss rate*.
- With multi-objective approach focused on Pareto-dominance, the quality of the whole approximation, in a Pareto dominance sense, highly depends on the worst approximated objective for a given solution making harder to extrapolate results on an objective taken individually.

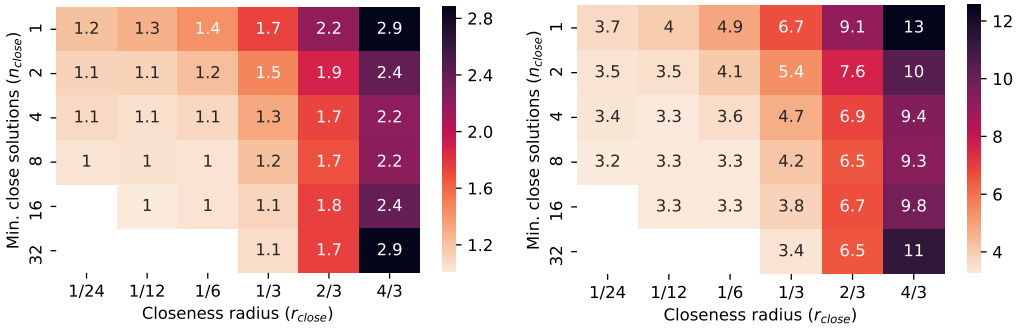
Due to the fact that there is no clear literature covering clearly all these points, we propose to keep the methodology simple enough by evaluating first the impact of the various parameters of each approach on the accuracy of PDM and ITDM utilities independently. Once the parameters chosen by this way, we complete with some additional experiments to provide some discussion on the limits and inherent issues of this problem.

Independent evaluation

To obtain a realistic dataset of power profiles and objective values, the individuals generated during SPEA2 runs are recorded and stored. These profiles were obtained using SPEA2 with the settings presented in section 5.5.1 (80 time steps with *Normal* scenario), 1600 evaluations and the chosen values for pb_{cross} and pb_{indmut}. This is repeated for 10 different runs, leading to a set of 8000 known profiles for both ITDM and PDM utilities.

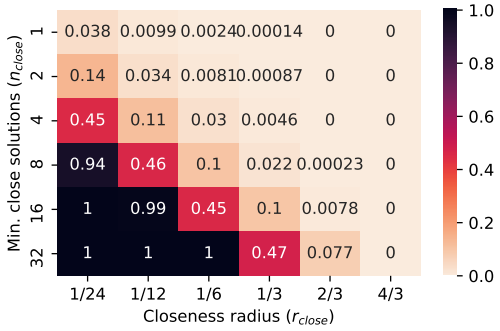
Using this dataset, each surrogate method is evaluated with different parameters. A subset of 100 profiles are randomly selected in the dataset to form the set of *known solutions*, while all the others are used to perform approximations and compare to the real utility. For AD and MHT, the number of these profiles for which the approximation algorithm is not successful is counted as *missed*. This process is repeated 10 times for each combination of parameters in order to have robust results. Two metrics are obtained by averaging results for a combination of parameters: the RMSD between successful approximation and real utility value of a profile, and the ratio of missed approximations for AD and MHT.

MHT and AD Our two surrogate methods are both controlled by two parameters with similar meaning, r_{close} and n_{close}. The value of each parameter is selected among 6 possibilities, following a geometrical progression. Values of n_{close} varies from 1 to 32 for both MHT and AD. The distance metric is however not the same between the two approaches: it is a derivative of Euclidean distance for MHT while it is a mean square error (MSE) for AD, which grows much faster than Euclidean distance. Different ranges of values are therefore studied for r_{close}, both obtained as geometric series of ratio 2, from



(a) RMSD with PDM utility

(b) RMSD with ITDM utility



(c) Approximation miss rate

Figure 5.9 – Miss rate and root-mean-square deviation (RMSD) for the Multiresolution Haar Transform (MHT) approximation method using 100 known profiles.

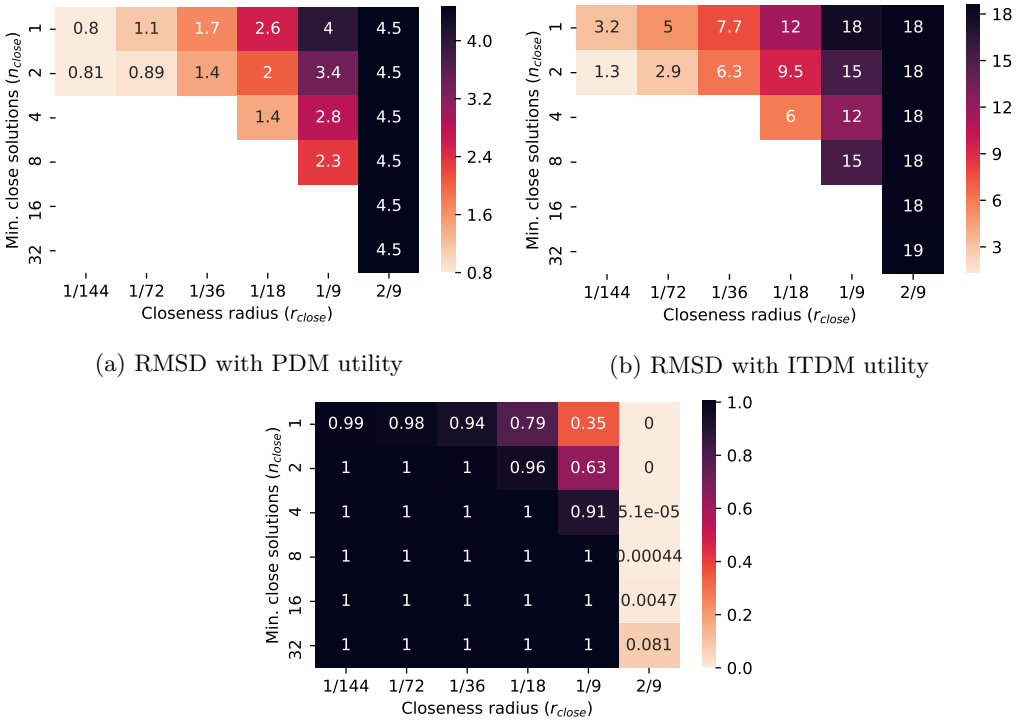
1/24 to 4/3 for MHT and from 1/144 to 2/9 for AD. Their start terms may appear somehow arbitrary for historical reasons, but are capturing accurately the different behaviors of the surrogate methods.

Figure 5.9 shows the results for the Multiresolution Haar Transform (MHT) approach. As the miss rates are the same when approximation is performed for either ITDM utility or PDM utility, they are given in a single figure 5.9c. Small values of r_{close} and/or large values of n_{close} lead to very important miss rates (up to 100%). This is expected, as both cause the requirement for performing the approximation to be more difficult to meet (closer solutions required for large n_{close} and smaller distance to known solutions for small r_{close}).

The quality of approximations is presented in fig. 5.9a for the PDM utility. As only the successful (*i.e.*, not missed) approximations are considered, it must be highlighted that combinations with miss rates close to 1.0 may have their quality computed with a very small number of samples. The worst values are obtained for large values of r_{close} (≥ 4) and for $n_{\text{close}} = 1$. While the scale of utility values (and therefore the RMSD) are not the same, the approximation of ITDM utilities gives very similar patterns, as shown in fig. 5.9b.

Ideal values for n_{close} and r_{close} would perform an approximation of all the profiles matching exactly the real utility value for this power profile (both miss rate and RMSD as close to 0 as possible). As none of the tested value combinations is perfect, a trade-off must be chosen. The combination selected is the best one for quality with a constraint on miss rate $\leq 3\%$, resulting for the MHT method in $n_{\text{close}} = 4$ and $r_{\text{close}} = \frac{1}{6}$.

For Average Distance (AD) approximation method, similar results are summarized



(a) RMSD with PDM utility

(b) RMSD with ITDM utility

(c) Approximation miss rate

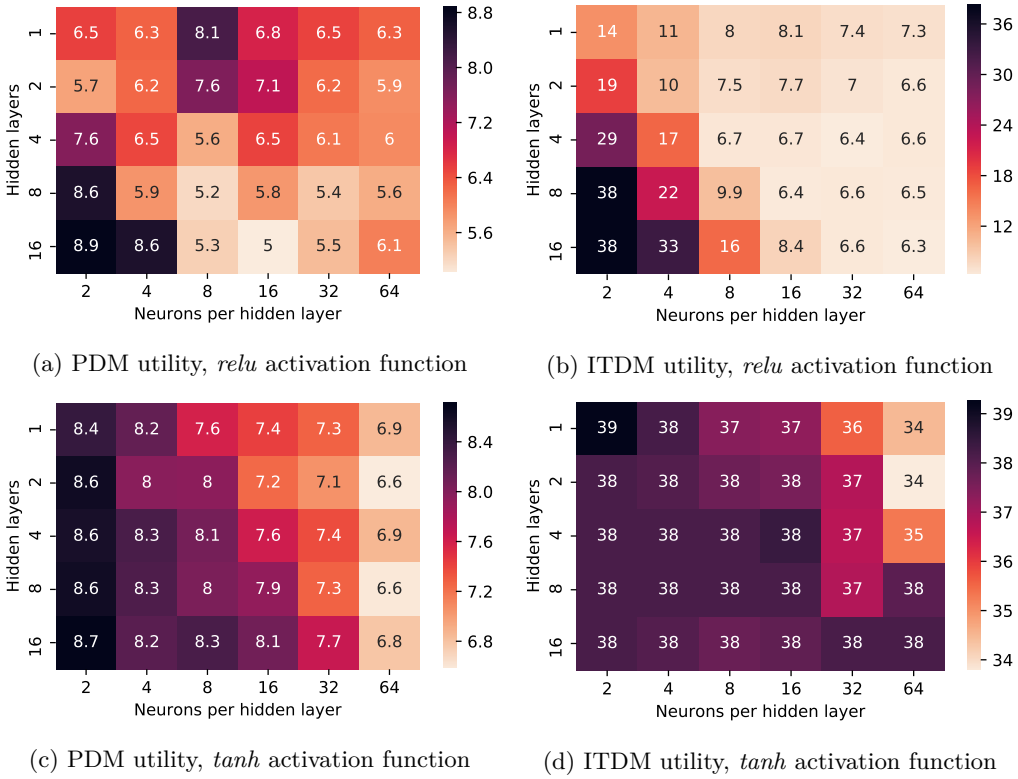
Figure 5.10 – Miss rate and root-mean-square deviation (RMSD) for the Average Distance (AD) approximation method using 100 known profiles.

similarly in fig. 5.10. The resulting miss rates shown in fig. 5.10c differ a lot compared to MHT results. This is expected as the distance metric used by both methods are not the same. Low miss rates ($\leq 5\%$) are only achieved with the highest value of r_{close}. A few other combinations result in miss rates lower than 90%, by requiring few close solutions to return an approximation ($n_{\text{close}} \leq 2$). Figures 5.10a and 5.10b present the quality of successful approximations for PDM and ITDM utility functions respectively. Similarly to MHT results, the RMSD is lower for small values of r_{close} and n_{close}, with a more important deviation for $r_{\text{close}} = 8$. However, the values suggest a worse approximation quality for all parameter combinations compared to MHT ones, except for some values which lead to very high miss rates ($\geq 99\%$).

Contrary to MHT results, where it was easy to find values with both a relatively good approximation quality and a low miss rate, no such combination appears for AD. As it seems important to have a relatively good approximation quality, a different trade-off is used to choose the parameters. While leading to a high (80%) miss rate, parameters $n_{\text{close}} = 1$ and $r_{\text{close}} = \frac{1}{18}$ lead to a relatively good approximation quality.

Multilayer perceptron hyperparemeters Neural networks in general are known to have many hyperparameters that are difficult to setup without testing many configurations for a given problem. Here, we focus only on a subset of them: the activation function and the shape of the hidden layers. The solver L-BFGS [125] showed better and more stable results in preliminary experiments compared to stochastic methods such as Adam and is therefore used in the presented experiments.

To study their impact, we proceed similarly than for the previous analysis of AD and



(a) PDM utility, *relu* activation function

(b) ITDM utility, *relu* activation function

(c) PDM utility, *tanh* activation function

(d) ITDM utility, *tanh* activation function

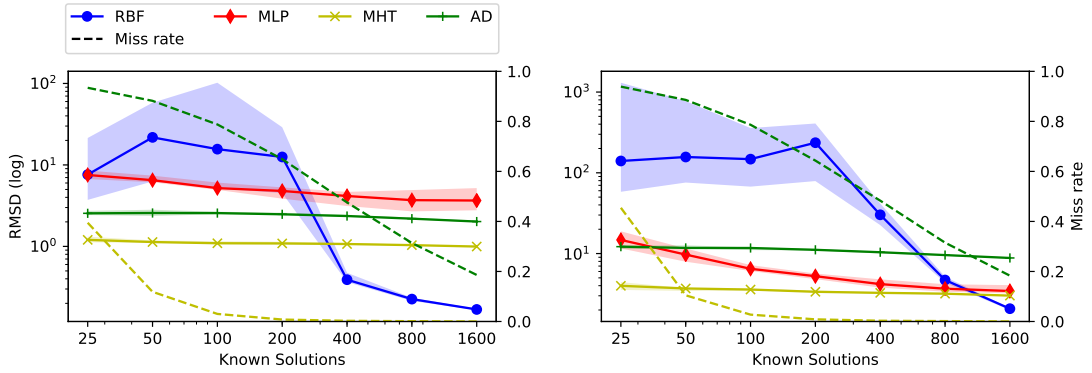
Figure 5.11 – Root-mean-square deviation (RMSD) for different hidden layers and activation function of Multilayer Perceptron (MLP), with 100 known profiles.

MHT methods, without the need to care about approximation misses as an MLP gives an approximation as long as the network was trained. The results of two activation functions are detailed, *relu* and *tanh*. Additional experiments with a *logistic* activation were made and lead to results slightly different but with many similarities with *tanh*. The structure of the hidden layers is studied by setting the number of layers (between 1 and 16) and the number of neurons on each layer (between 1 and 32).

Figure 5.11 presents the results for both activation function, showing RMSD for PDM utility on the left and for ITDM utility on the right. For recall, the scale of ITDM utility values being higher than PDM ones on the studied profiles, higher RMSD are expected for the first.

We clearly see that *tanh* perform badly on this problem, with results on ITDM utility on fig. 5.11d being significantly worst than other surrogate methods for all tested parameters. Its accuracy for PDM utility, detailed in fig. 5.11c, is comparatively better but stays behind the ones of *relu*. For *relu* activation function, some better results are observed especially for ITDM utility in fig. 5.11b. The absolute best accuracy is obtained for maximum values of parameters (64 neurons per hidden layer and 16 layers), but close values are obtained by lowering this high number of layers. For the PDM utility, shown in fig. 5.11a, the better values are reached with 8 or more neurons per hidden layer and 8 or more layers.

A set of parameters which leads to a good trade-off between ITDM and PDM accuracy is using *relu* activation with 8 layers of 32 neurons. These figures seem really high considering our problem, especially the number of layers as many works suggest that 2 or 3 hidden layers is generally sufficient for approximating any function. However, a small training set is rarely considered, with in our case only slightly more samples than input



(a) PDM utility

(b) ITDM utility

Figure 5.12 – Median and spreading (top and bottom deciles) of the root-mean-square deviation of the different selected surrogate, depending on the number of samples of the real utility values (known profiles). Miss rates of MHT and AD are given on the right axis, as dashed lines.

neurons. While the values found are surprising, the methodology used is clear and fair with AD and MHT methods. Therefore, these parameters will be used for MLP in the rest of the experiments.

Impact of number of known solutions on accuracy To help understanding better the behavior of the surrogate methods, an additional set of experiments is performed using a similar setup. Using the selected parameters for each approach, the accuracy and miss rates are studied for different sizes of known solution pool (training set). Similarly to the previous experiments, a number of solutions are selected randomly in the large pool of 8000 profiles and the rest is used for evaluation. Each combination of surrogate method and number of known solutions leads to 10 experiments with different training set.

Results are given in fig. 5.12 for both ITDM and PDM utilities, as log-log plot and with the spreading displayed as a surface around the median value. It shows first that MHT and AD only slightly improve their accuracy when more solutions are known. However, their RMSD is quite good compared to the other methods for few known solutions, especially MHT which outperforms all the other methods for up to 200 samples for PDM utility and 800 for ITDM one. By looking at their miss rates, it seems to confirm that the concept of approximation miss helps to keep the accuracy good when training set is small.

The RBF method exhibit unique behaviors among the different surrogate methods. The first noticeable one is the huge spreading for low number of known solutions (≤ 200). This is a good indication that it is not adapted for such conditions, being highly unstable as small changes on the data may cause high variations of the radial function weights. The second observation is its very high accuracy for large budgets (≥ 400 for PDM and ≥ 1600 for ITDM utility). Past this amount, it largely outperforms the other methods and behave in a stable way.

Finally, the multilayer perceptron method gives unequal results on PDM and ITDM utilities. For the PDM utility, it slightly improves with the number of known solutions, but becomes more and more unstable and never performs better than AD or MHT. This may be a consequence of a badly adapted method for this problem combined with some overfitting phenomenon [172] (neural network trying to fit "too well" the set of known solution and resulting in inaccurate approximation for other inputs).

While these results are helpful to understand the surrogate model themselves, for the reasons mentioned previously they are not directly usable to predict the behavior of the overall optimization process.

Multi-objective joint approximations

While they are not used to choose the approximation methods parameters, additional experiments are performed to highlight the behaviors of these surrogate models when used simultaneously for multiple objectives. Two cases are studied for the use of surrogate models to approximate both ITDM and PDM utilities: independently from the MOEA like the previous section and during the optimization. In the following, both the ITDM and the PDM approximation functions are set with the same method and same parameters.

As our approach is based on Pareto strength of individuals, it is important that the surrogate models predict accurately the Pareto dominance of individuals over the known solution set. If a new individual is considered, by the approximation functions, as dominated by the current set of best individuals, it would very likely be eliminated before having a chance of being evaluated with the real utility functions. It must be highlighted that a single *pessimistic* approximation function is enough for mis-considering a dominating solution. This makes the whole approximation quality depending mostly on the worst of the approximation functions.

To characterize more precisely the combined use of surrogate models (one instance for ITDM utility and another for PDM utility), a set of 5 indicators are measured. All of them represent a percentage of the approximations and are to be maximized for an ideal surrogate model.

- *hit rate*, part of successful approximations (not leading to a *miss*)
- *true dominated*, part of individuals dominated by the set of known solutions (with the real objective values) which are also considered as dominated by the surrogate model
- *true non-dominated*, part of individuals which are not dominated by the set of known solutions correctly considered as non-dominated by the surrogate model
- *miss desired*, part of non-dominated individuals among the missed approximations and would therefore cause a useful evaluation (set at 100% for RBF and MLP)
- *overall desired*, part of individuals which are either *true dominated*, *true non-dominated* or *miss desired* among all the performed approximations

A first experiment series is done independently from the real optimization loop, using a pool of known solutions and without MOEA. Because this allows to study the behavior of the approximation method before using them in an optimization framework, it is called here *a priori* evaluation. Using the same settings as in section 5.5.2, a set of 100 pairs of solutions and their real objective value are chosen for becoming the set of known profiles for both ITDM and PDM approximation functions. All the other solutions are used to approximate their objective values, which is compared to the real ones.

Figure 5.13 summarizes the different indicators for the 4 surrogates using the chosen parameters. The values of hit rate and miss desired are only interesting for the approaches handling approximation misses, MHT and AD. The miss rates were already studied for selecting the parameters of both approaches and the low hit rate for AD is not surprising.

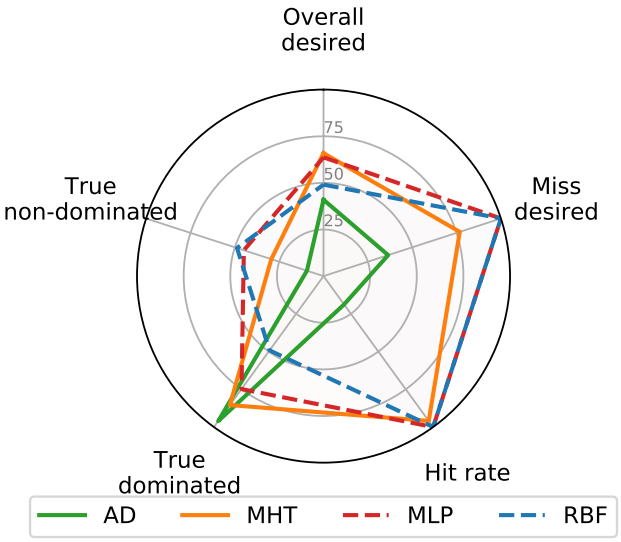


Figure 5.13 – Pareto-dominance related performance of surrogate methods, based on random samples of known profiles and their utilities.

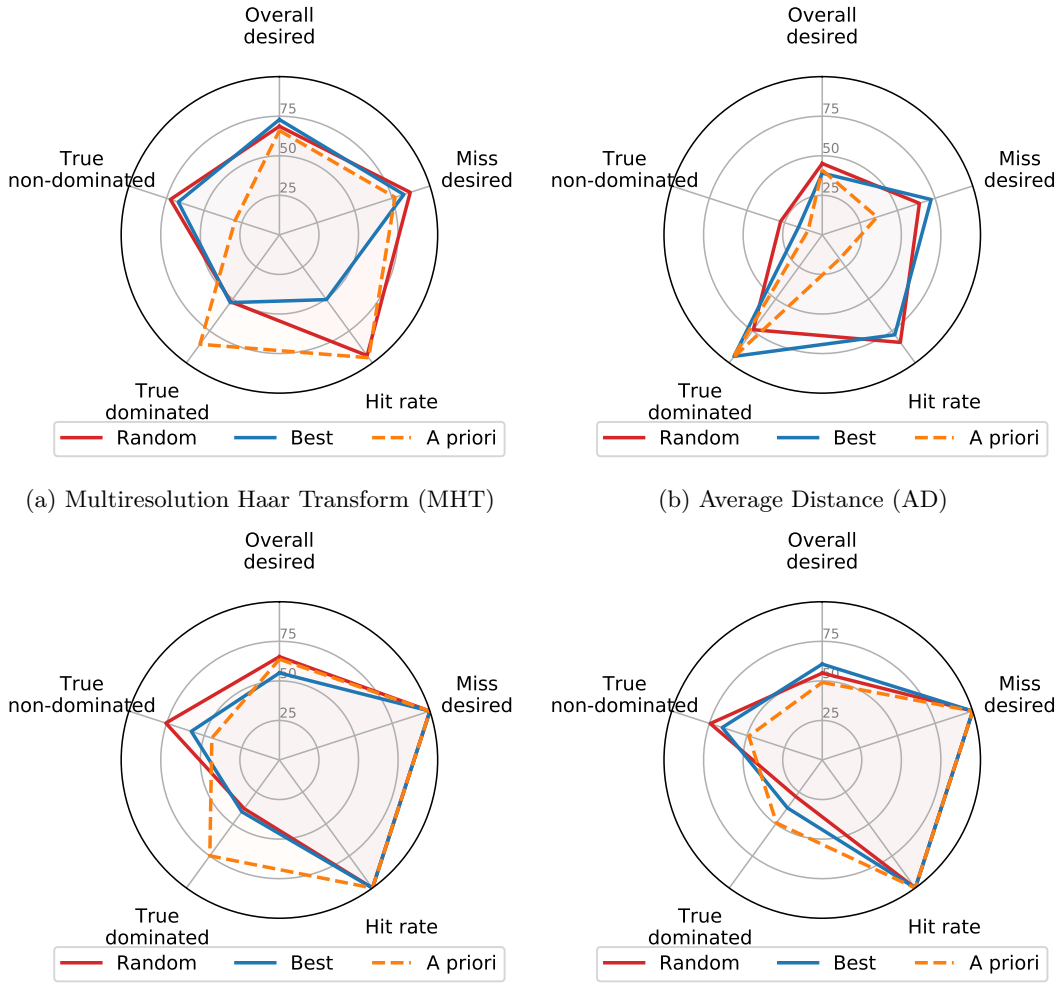
The part of prediction miss which led to a non-dominated solution (miss desired metric) is higher (77%) for MHT than for AD (36%).

Interestingly, very different behaviors are observed on the true dominated and true non-dominated metrics. MLP and RBF performs similarly to correctly predict a new solution dominating the known ones (true non-dominated), with almost 50% of correct predictions (45% and 49% respectively). The AD method is the worst for this metric (9.3%) and MHT is in-between, with 29% of correct guesses. The results are almost inverted for true dominated ratio, with AD predicting correctly 96% of dominated individuals, followed by MHT (85%) and MLP (75%). The radial-basis function surrogate is the last, with just below the half of them (49%) correctly considered dominated by known solution set. These results show that AD is extremely "pessimistic", but avoid considering potentially interesting solutions which would have not lead to any improvement of the current Pareto-dominating solutions. At the contrary, MLP appears quite balanced in terms of Pareto-accuracy of predictions.

In this *a priori* evaluation, despite the observed accuracy on independent objectives, the overall rate of desired outcomes is similar between MHT (66%) and multilayer perceptron method (64%). RBF does slightly worst (49%) and AD is still getting an acceptable value of 41%, probably because of the high figure of true dominated indicator.

In the second experiment series, the same characteristics of the surrogate models are analyzed *in situ*, during a complete optimization run, using USPEA2 along with the approximation methods. While this is not practical for tuning the surrogate models for the overall approach, it is useful to understand how the overall MOEA and surrogates are working together. The parameters used for USPEA2 and for the ITDM and PDM models are the same as in experiments presented later in section 5.6.4, using scenario *Normal* with 80 time steps. The end condition is set when 100 evaluations are performed by the real objective functions ($N = 100$) and each experiment is repeated 10 times in order to average the results. Two different cases are compared. One is using the *Random* initial population scheme, the other is using the *Best* initial population (one individual considered as the best solution for each ITDM and PDM, completed with random individuals).

Figure 5.14 presents the indicators for each approach, along with their results in *a*



(a) Multiresolution Haar Transform (MHT)

(b) Average Distance (AD)

(c) Multilayer Peceptron (MLP)

(d) Radial-Basis Functions (RBF)

Figure 5.14 – Pareto-dominance related performance collected during USPEA2 optimization and using either a random initial population (*Random*) or including the best profiles of each decision module (*Best*). Results for *a priori* evaluation from fig. 5.13 is given for comparison.

priori setup. A first observation concerns the differences with the *a priori* analysis. Some characteristics are similar, such as the high values of true dominated indicator for AD. It is also notable that despite behaving differently, the overall desired metric is very stable and close in all condition to the value obtained *a priori* (less than 10% of absolute difference). However it seems that the values observed for true non-dominated are systematically higher *in situ* and systematically lower for true dominated ratio. The hit rate is also significantly higher for AD (between 78% and 84% instead of 19%), as well as for the miss desired ratio.

The indicator values for each initial population settings are close (less than 15% absolute difference) in most cases. A major exception concerns the hit rate of MHT, which is around 94% when initial population is *Random*, but as low as 51% with the *Best* scheme. Such a difference may be explained by the relatively high number of close known solutions ($n_{\text{close}} = 4$) required for performing an approximation. As the best solutions of both decision modules are likely very different, several areas of the solution space, far from

each other, are explored early in the optimization process. This makes less probable to known already enough solutions close enough to a new individual obtained after crossover of solutions far from each other.

All these results together are very useful to understand two main points. As stated previously, the surrogate models and the MOEA behaviors are closely tied together. This is highlighted by the differences observed for some indicators between the *a priori* analysis (which attempts to isolate the approximation functions from the MOEA) and the *in situ* one (studying both used together). While some differences may be explained by not using a good enough solution sample set in the *a priori* method, some others, such as the hit rates of AD, are more likely explained by the interactions between the MOEA and the surrogate models. The second point is the impact of initial population over the performance of the approximation functions. The different behaviors observed in the two settings give a good reason to expect different performances for the overall MOEA and surrogate models, depending on the initial population.

5.5.3 Lifetime of approximation

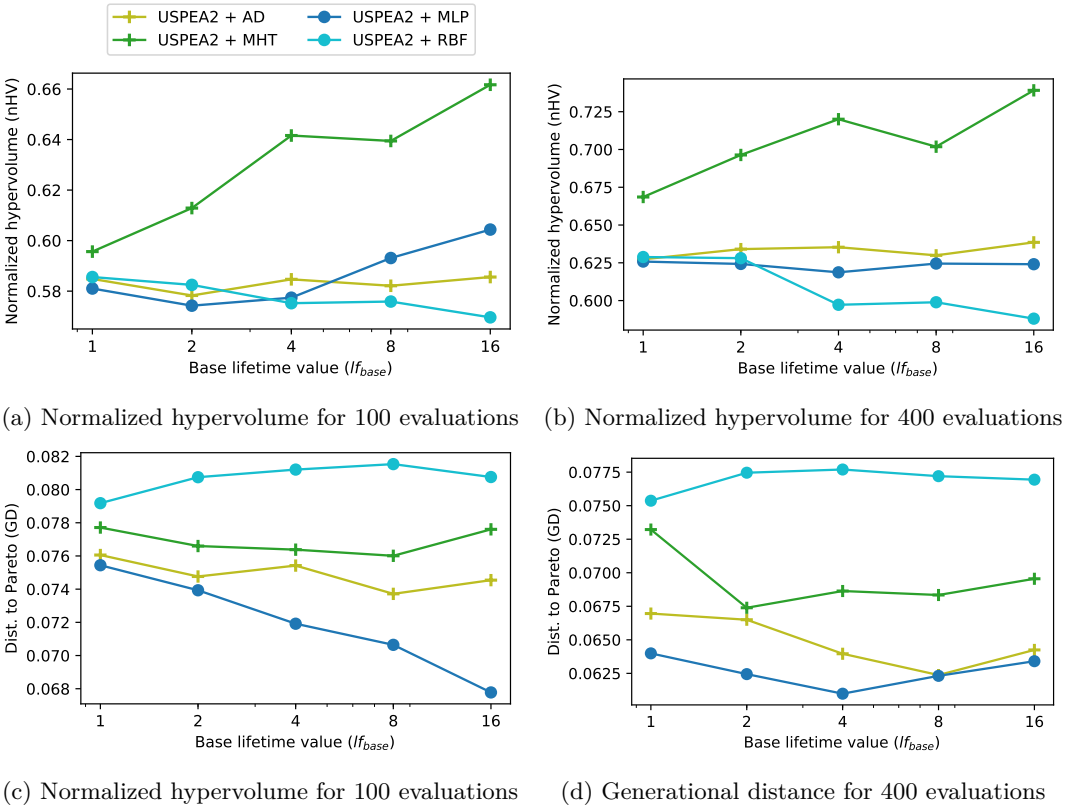
The last unset parameter of the approach is lf_{base} (initial lifetime of approximated objectives) which is meaningful only when combining both optimization and approximation algorithms. Using the previously detailed values for all the other parameters, USPEA2 with the approximation algorithms are evaluated with end condition set to either 100 or 400 evaluations. The initial population is generated using the *Random* scheme. The experiments are performed 10 times for each settings. A total of 4 values of lf_{base} are chosen between 1 and 16, following a geometrical progression.

Higher values of lf_{base} mean that an individual with approximated objective functions needs to survive more generations in order to be evaluated with the real objective functions. While this allows to reduce the amount of required evaluations, it also increases the risk of keeping bad individuals longer.

Figure 5.15 shows the resulting normalized hypervolume (to maximize) and generational distance (to minimize) of the final solution set for each approach. For a low target of objective functions evaluations ($eval_{max} = 100$), approaches either do not improve significantly (AD and RBF) or give better results with higher lf_{base}, as shown by figs. 5.15a and 5.15c. This is especially visible for MHT, for which the resulting nHV increases from 0.596 for $lf_{base} = 1$ to 0.661 for $lf_{base} = 16$. Similarly for multilayer perceptrons surrogate, the average distance to Pareto front (GD) is continuously decreasing down to 0.068.

Figures 5.15b and 5.15d present the results for a higher evaluation budget ($eval_{max} = 400$). In this case, the benefit of a high value of lf_{base} is less clear, with only MHT showing clear improvement on hypervolume, with a small decrease at $lf_{base} = 8$. It is however not significant according to its standard deviation (0.045 for this value of lifetime). Contrary to the results obtained with a lower budget, the generational distance is almost not improved for $lf_{base} \geq 2$ and is even slightly impacted negatively for $lf_{base} \geq 8$.

Using a value of $lf_{base} \geq 8$ appears to give good results when USPEA2 is used. However, increasing the value of lf_{base} also tends to increase the number of generations of the genetic algorithm required to reach a solution, as more generations are needed for an approximated individual to be evaluated. Even if the execution time is expected to be bound by the evaluation of real objective function themselves, the initial lifetime of approximated individuals should not be chosen uselessly too high. Increasing lf_{base} from 8 to 16 present an improvement for $eval_{max} = 400$ only possibly for MHT and also for MLP for lower budget ($eval_{max} = 100$). Therefore, the value used in the rest of the experiments presented here is $lf_{base} = 8$.



(a) Normalized hypervolume for 100 evaluations

(b) Normalized hypervolume for 400 evaluations

(c) Normalized hypervolume for 100 evaluations

(d) Generational distance for 400 evaluations

Figure 5.15 – Impact of the base lifetime (lf_{base}) value on hypervolume and generational distance of resulting Pareto approximation when using the different surrogate methods.

Symbol	Meaning	Value
pb_{cross}	Probability of crossover	0.5
pb_{indmut}	Probability of a timestep to change during a mutation	0.072
archiveSize	Maximum size of the archives	20
populationSize	Maximum size of the population	20

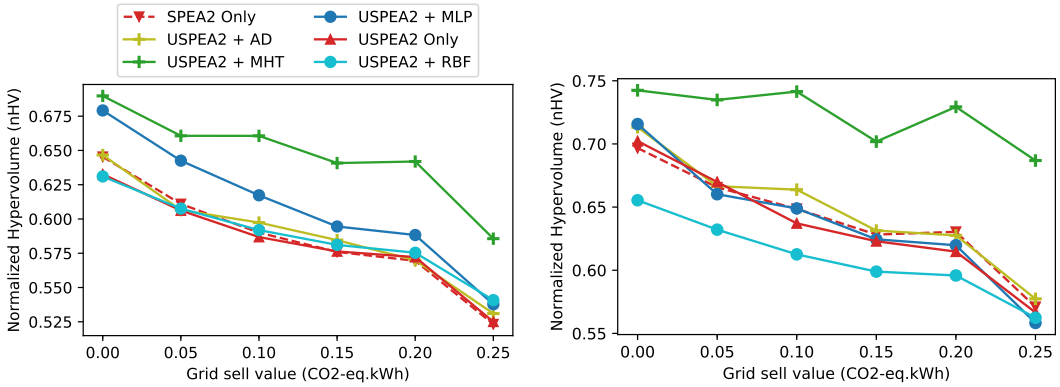
Table 5.3 – Parameters used for SPEA2 and USPEA2.

5.6 Experimental results

In order to validate the proposed algorithms, several series of experiments are conducted, using the parameters defined along the previous section and summarized in tables 5.3 and 5.4. To evaluate the robustness of the algorithms with respect to the objective functions definition, the impact of an important constant of the PDM utility function on the quality of the optimization is studied first. Then, the behavior of each heuristic depending on the initial population used is studied. This shows how different level of initial knowledge of the solution space may impact the quality of the result for each heuristic. Finally the results are compared for all the defined scenarios and different sizes of the solution space (number of steps in the time window, T) to show the robustness of the heuristics depending on the problem definition and how they scale for huge solution spaces.

Method	Parameter	Value
MHT	r_{close}	1/6
	n_{close}	4
AD	r_{close}	1/18
	n_{close}	1
MLP	Neurons per layer	32
	Hidden layers	8
	Activation function	*relu*
RBF	Radial function	*cubic*

Table 5.4 – Parameters used for each objective function surrogate method.



(a) Normalized hypervolume with 100 evaluations. (b) Normalized hypervolume with 400 evaluations.

Figure 5.16 – Impact of the grid sell utility parameter (ghg_{sell}) for the different approximation methods used with USPEA2.

5.6.1 Impact of PDM utility formulation

The quality of the objective function approximation may depend on various characteristics of the real utility functions themselves. For instance, functions with clear non-linearity are usually more challenging to approximate, whereas very symmetrical ones may instead lead to better approximations. On the other hand, the formulation of utility functions also impacts directly the shape of objective space of feasible solutions. This may lead to larger or sparse space, more difficult to explore with MOEA in general.

In the PDM utility as formulated in section 5.4.1, the added utility for selling energy on the grid (ghg_{sell}) causes some kind of symmetry, depending on its value, compared to the utility removed for buying on the grid (ghg_{buy}). For instance, when $ghg_{sell} = ghg_{buy}$, any profile with the same amount of energy ($\sum_{t=0}^{T-1} p_t \cdot \delta$) will have the same PDM utility as the renewable energy may be sold directly to the grid when produced and bought at anytime with no impact on resulting utility. However this also extends the range of PDM utility values and consequently the size of the real Pareto front. By looking at the behavior of the approaches for different values of ghg_{sell}, it is possible to have an overview of the impact of such utility function property on the overall approaches. Different values are used, from 0 (selling energy for no benefit) to ghg_{sell} (selling a unit of energy compensates exactly the purchase cost).

Figure 5.16 gives the resulting normalized hypervolume for each value of ghg_{sell}, with end condition set to a budget of $eval_{max} = 100$ or $eval_{max} = 400$ evaluations of real objective functions. A general trend for all the heuristics is that values closer to ghg_{buy} result in a lower hypervolume, likely because of the increased objective space. The behaviors of each surrogate differs between the two budgets of evaluations. Particularly, for $eval_{max} = 100$ MLP gives results higher than the baseline SPEA2 without surrogate with $ghg_{sell} = 0$ and becomes closer with increased values. With higher budget, using multilayer percetrons surrogate leads to no significant improvement of hypervolume. Similarly, USPEA2 with radial-basis functions (RBF) is very close to the baseline for low budget, but results in smaller values for $eval_{max} = 400$.

The proposed Multiresolution Haar Transform approach gives higher hypervolume in both cases. However, for a low evaluation budget, the relative improvement compared to SPEA2 without surrogate method is similar for all values of ghg_{sell}, whereas it increases with it when $eval_{max} = 400$. Te other approaches, USPEA2 without surrogate and with AD, perform similarly to the baseline.

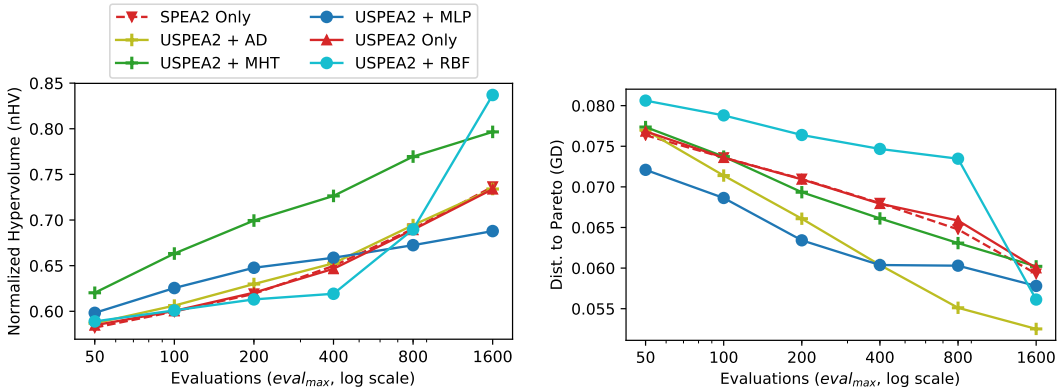
5.6.2 Maximum number of evaluations

The problem considered here is a black-boxes optimization with evaluation budget constraint. The more evaluations are allowed, the more resource and time are required to perform the optimization. To be able to take a decision in a reasonable amount of time, the main target is between 100 and 400 evaluations. To understand how this budget affects the quality of the resulting Pareto front approximation, a larger range is evaluated.

The results of each variant are compared for $eval_{max}$ between 50 and 1600 in fig. 5.17, with scenario *Normal* and 80 time steps in the time window. The variant *SPEA2 + None* (original SPEA2 algorithm without approximation method) is considered as the baseline for comparing the other results. The objective values of each individual from the initial population must be either evaluated or approximated using the surrogate model in the same way as any further new individual. Therefore, it must be highlighted that for small values of $eval_{max}$, a significant amount of evaluations may be spent *before* the first series of crossover and mutations. It may be up to $2 \cdot populationSize$ (one costly evaluation per decision module), or 40 evaluations in case none of them are successfully approximated.

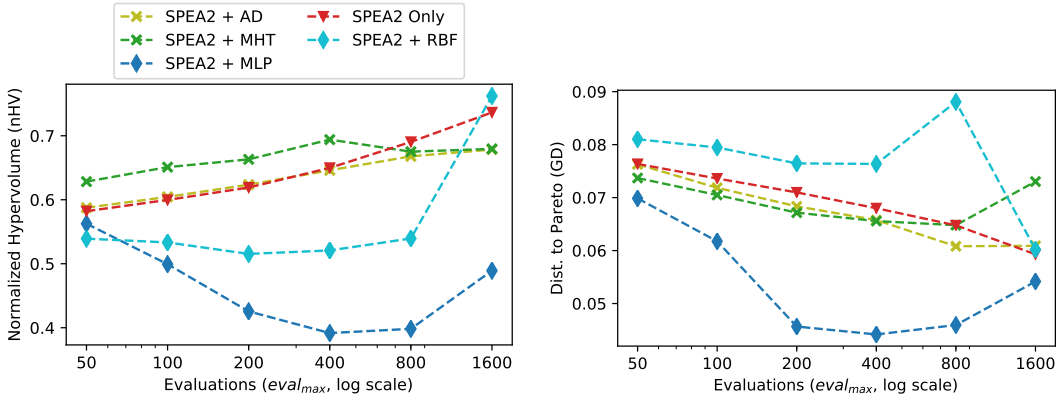
For all the methods using USPEA2, the hypervolume increases with higher number of evaluations, as shown in fig. 5.17a. The relationship between hypervolume and $eval_{max}$ is almost logarithmic for all of them except RBF, which increases quickly when $eval_{max}$ exceeds 400. The approximation method MHT lead to the highest hypervolume for all the values of $eval_{max}$ below the maximum tested of 1600, in which RBF gives slightly higher result. The relative improvement of MHT on hypervolume compared to the baseline is between 6.5% (for $eval_{max} = 50$) and 13% (for $eval_{max} = 200$). The other approaches are close to the baseline. The MLP surrogate methods, initially slightly upper the baseline, crosses it after $eval_{max} = 400$ and ends up 6.6% below for the highest budget. For all the tested conditions, USPEA2 and SPEA2 without surrogate are giving similar values, with less than 0.5% relative difference.

Figure 5.17b shows the average distance to Pareto, or generational distance, which must be minimized. The global trend is a logarithmic decrease with the evaluation budget, with RBF diverging here too for $eval_{max} = 1600$. The Average-Distance (AD) and MLP surrogate methods result in lower generational distances than the baseline. The gap is increasing with $eval_{max}$ for AD and globally decreasing for MLP. On this metric, MHT is always close to the baseline (maximum relative difference of 3%). Without using surrogate methods, USPEA2 and SPEA2 result in very close generational distance (difference $< 2\%$).



(a) Normalized hypervolume

(b) Average distance to Pareto front



(c) Normalized hypervolume

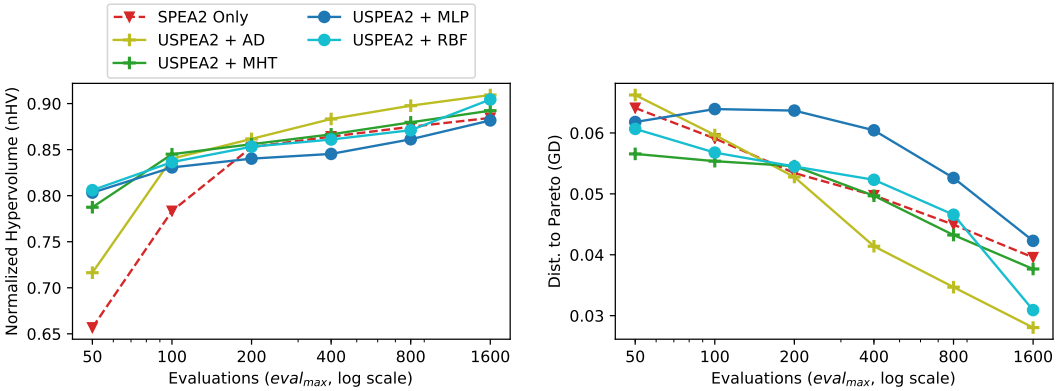
(d) Average distance to Pareto front

Figure 5.17 – Performance of each variants for different evaluation budgets, with *random* initial individuals.

Results when using SPEA2 together with the different surrogates are presented in fig. 5.17c for hypervolume and fig. 5.17d for generational distance. Their behaviors are different from the ones observed with USPEA2, especially for high budgets ($eval_{max} \geq 400$). For all the values of $eval_{max}$, the hypervolume of the Pareto front approximation obtained is lower than with the USPEA2 algorithm, with the most noticeable degradation observed for MLP and RBF. The generational distance is however lower in some conditions, with a significant gap for MLP but also slightly better results for MHT. The explanation of this phenomenon, counter intuitive at first, is discussed in the next section. As similar results are observed in all other experiments, we focus only on USPEA2 when surrogate models are used in the rest of the section. With the same argument, SPEA2 and USPEA2 with no objective approximation method giving comparable results in all the other conditions studied (less than 2.5% of difference), only SPEA2 is given for comparison.

5.6.3 Initial population schemes

In genetic algorithms, the choice of initial individuals may greatly impact the convergence speed. As explained in section 5.4.2, an alternative way to generate the initial population is proposed in addition to the fully *Random* one. The *Best* scheme includes an optimal solution of each decision module, which maximizes its internal utility without other considerations, completed by random individuals to reach the size of the population. As this



(a) Normalized hypervolume

(b) Average distance to Pareto front

Figure 5.18 – Evolution of the performance with increasing evaluation budget and *best* initial individuals.

individual is expected to be the optimal for one of the objective, it must be on the actual Pareto front.

The experiments here are identical to the ones from section 5.6.2 except for the initial population scheme used. Similarly, the normalized hypervolume and the generational distance of each variant are presented in fig. 5.18 for different values of $eval_{max}$.

Compared to the use of the *Random* scheme in fig. 5.17, all the methods give higher hypervolume and lower *GD*. Both metrics also improve with the number of costly evaluations allowed $eval_{max}$ for all the USPEA2-based variants. Contrary to what is observed for the *random* scheme, there are important differences between surrogate methods on resulting hypervolume for low number of evaluations, up to 23% of the baseline value for $eval_{max} = 50$. In this setup, RBF and MLP give both the best results for such low evaluation number, followed closely by MHT (+20%) and finally AD improves the baseline hypervolume by only 9%. This difference between surrogate methods is significantly lowered with higher budget of evaluations ($\pm 3\%$ for $eval_{max} = 1600$).

The generational distance shown in fig. 5.18b is also globally improved compared to *Random* initial population. The behaviors of MLP and RBF are however almost reversed, with MLP giving worst (higher) generational distance than SPEA2 without surrogate for $eval_{max} \geq 100$. The AD method still exhibits lower distance to Pareto front for moderate to high budgets ($eval_{max} \geq 400$).

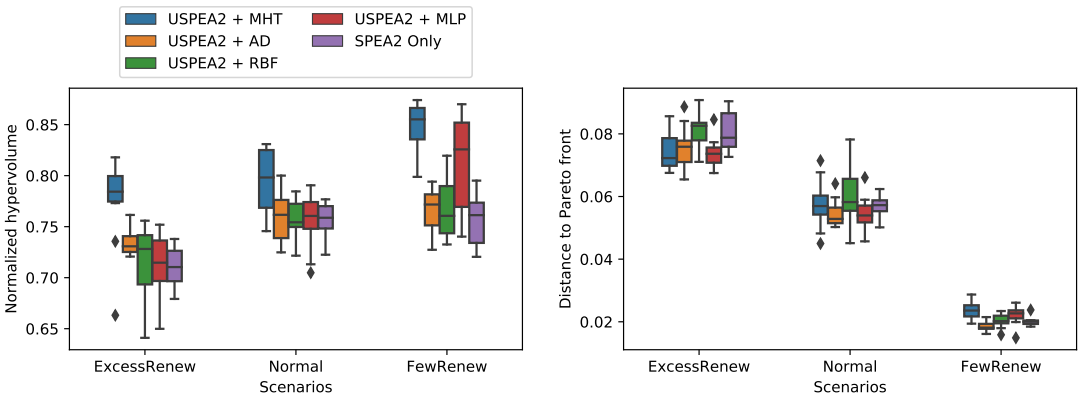
5.6.4 Scenarios and number of time steps

In order to evaluate the behavior of the approximation methods in various situations, experiments are performed for each power scenario defined in section 5.4.2. Another interesting point to show is the behavior of each approach when the size of problem change. As described in section 5.1, the number of time steps in a power profile (T) is also the number of dimensions of the solution space. Therefore, the same 72-hours time window is considered with either 20, 80 or 320 time steps. The two targeted evaluation budgets, 100 and 400 are also studied.

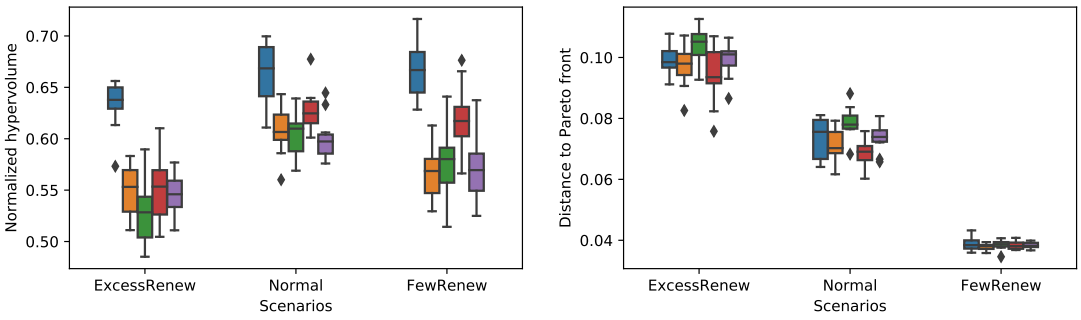
The results for $eval_{max} = 100$ evaluations are shown in fig. 5.19. For all the algorithms, better results are always achieved for lower number of time steps, both for normalized hypervolume and generational distance. For both metrics, there is also a noticeable difference between each scenario, with *FewRenew* leading to better overall results and *ExcessRenew*

to slightly worst ones, especially for *GD*. With the exception of RBF in a few cases, the resulting *GD* for a given scenario and number of time steps is close between all the variants. The combination of USPEA2 with our proposed MHT surrogate gives the highest hypervolume in all the conditions, with a lower gap compared to the other in the *Normal* scenario. Using multilayer perceptrons (MLP) does not improve significantly the hypervolume except for the scenario *FewRenew* and for large time window ($T = 320$).

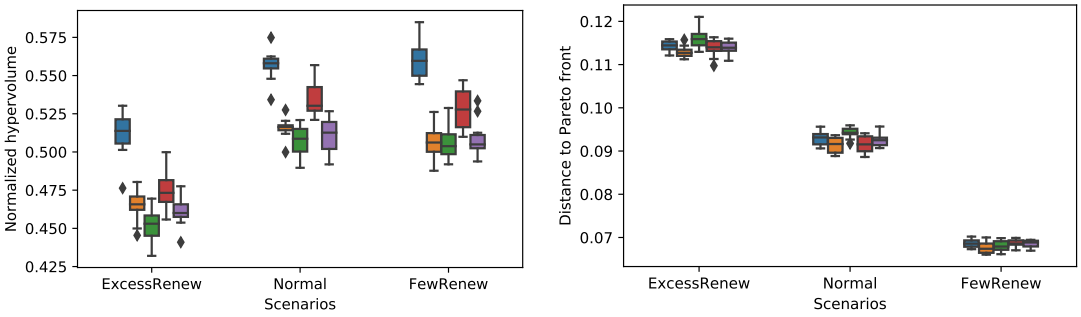
Figure 5.20 presents the results for a higher budget of costly evaluation $eval_{max} = 400$. The same observation on the global impact of the scenarios is still accurate for both generational distance and hypervolume. A major difference between the results detailed for lower budget is the behavior of the Radial-Basis Functions (RBF) surrogate. It leads



(a) Window of $T = 20$ time steps.

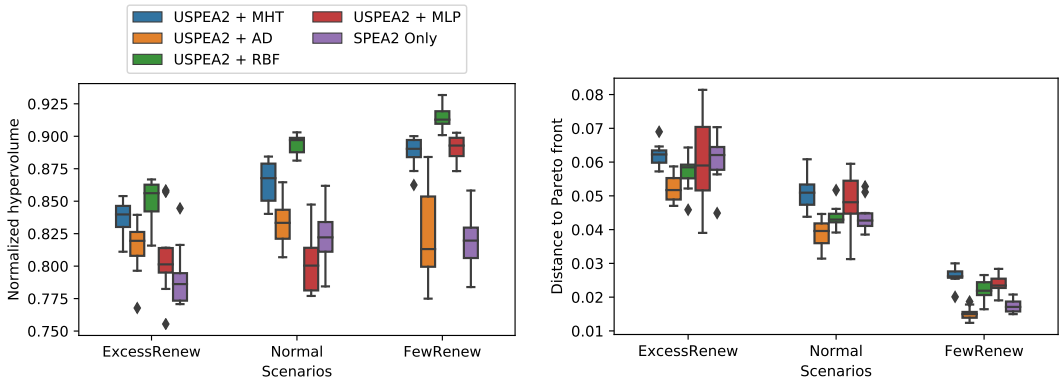


(b) Window of $T = 80$ time steps.

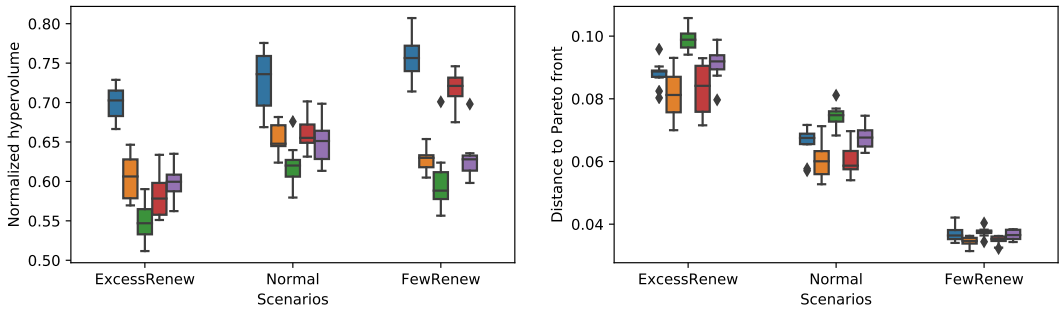


(c) Window of $T = 320$ time steps.

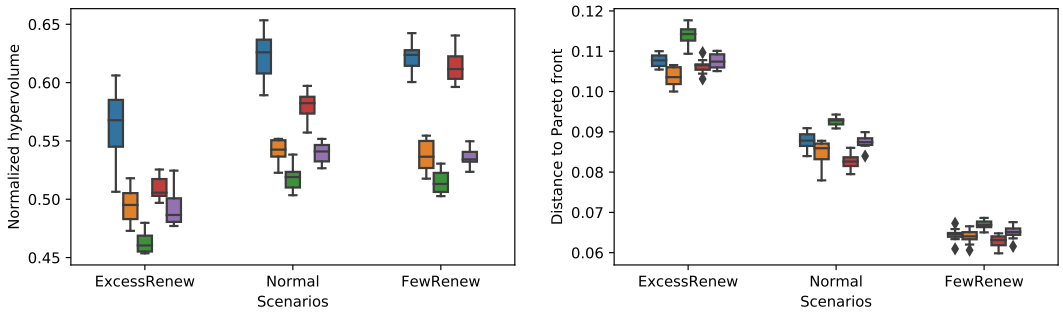
Figure 5.19 – Median and quartiles of both metrics (normalized hypervolume and generational distance) for the different heuristics with a budget $eval_{max} = 100$ costly evaluations over several scenarios.



(a) Window of $T = 20$ time steps.



(b) Window of $T = 80$ time steps.



(c) Window of $T = 320$ time steps.

Figure 5.20 – Median and quartiles of both metrics (normalized hypervolume and generational distance) for the different heuristics with a budget $eval_{max} = 400$ costly evaluations over several scenarios.

to the higher hypervolume values for all the scenarios when the number of time steps is low ($T = 20$) but inversely to worst results (both on hypervolume and generational distance) than the baseline SPEA2 for larger time windows. On all scenarios, MHT gives the highest hypervolume when medium or large time windows are considered and is below RBF for the short case $T = 20$. The generational distance of all approaches is close from the baseline in most conditions, with only AD giving systematically better values.

5.6.5 Using complex IT model

The previous experiments were performed on a simplified infrastructure model, with therefore objective functions exhibiting less complex behaviors. Results presented here are in-

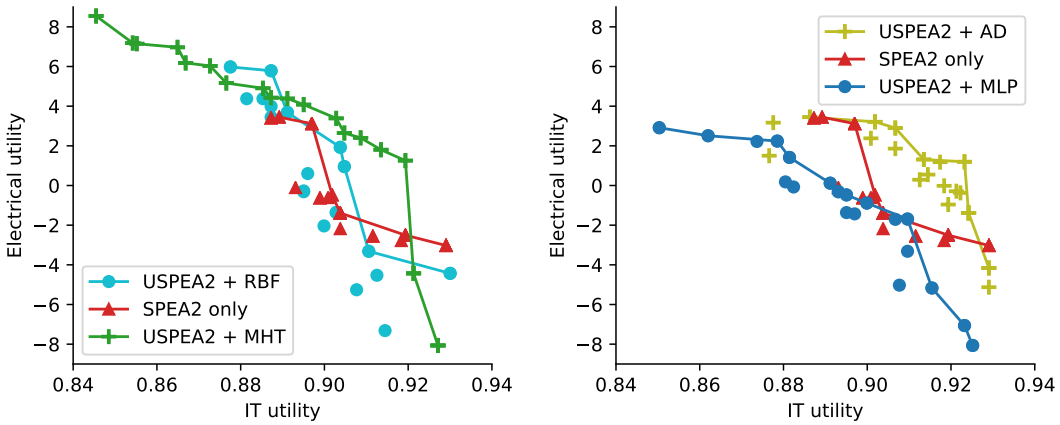


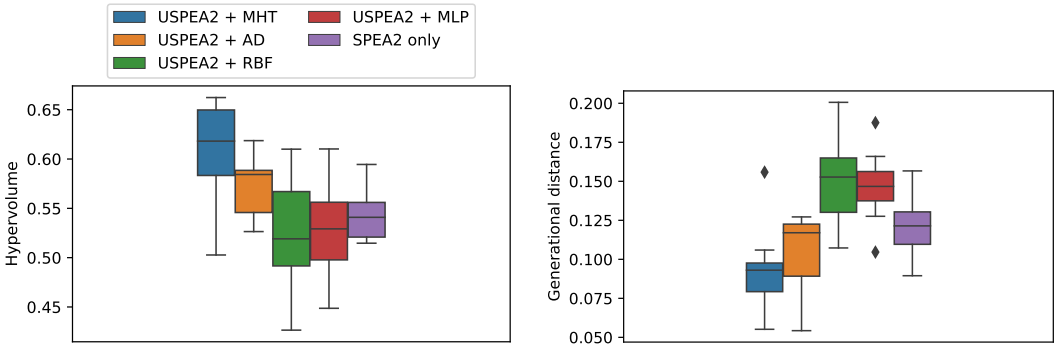
Figure 5.21 – Resulting solutions and Pareto front approximation after 100 costly evaluations, using RECO scheduler from the literature and complex IT infrastructure.

stead using the RECO scheduler [27] with a more realistic machine and tasks model. The experimental setup consists in the *Normal* scenario, with $T = 80$ and a budget of 100 costly evaluations.

Figure 5.21 shows the solutions obtained by running the different approaches over a complete optimization run. The approximated Pareto front from each approach is highlighted, but dominated solutions are also plotted as individual points outside the front. While this result alone does not allow a deep analysis, it provides valuable clues. On this run, the capacity of MHT and MLP to spread across the extremum is clearly visible, while AD is only improving a very local area towards the maximum values. MHT is the only method which ends with all its solutions being Pareto-optimal between them. At the contrary, AD only 9 out of 20 solutions fulfill this condition. It can also be noted that AD and MHT both results in most of their solutions dominating the ones of SPEA2 without surrogate method.

As mentioned earlier, the high complexity of this IT model prevent to obtain the real Pareto front of the problem. To allow comparing the approaches with higher level metrics, some modifications of the hypervolume indicator and generational distance are proposed. The hypervolume can still be computed but not normalized with the real Pareto front. To keep its value in meaningful range, it is calculated after normalizing each objective value by considering the minimum and maximum found among all the experiments. The generational distance requires a Pareto front to which measure the distances from each solution. While this is not commonly done in the literature, we propose for these experiments to extract the Pareto-optimal solutions among all the results to form the *best approximation* of the Pareto front $PF\star$. Generational distance is then computed as for the other experiments using this approximation instead of the MILP-based front.

These modified metrics are compared between the approaches in fig. 5.22, over 10 runs with the same parameters. Using the MHT surrogate method gives better results than the other approaches most of the time, for both hypervolume and generational distance metrics. On the hypervolume indicator, using any surrogate method leads to more spread results compared to SPEA2 alone. In addition, RBF and MLP appear to give worst results than the baseline at least half of the time. Finally, using USPEA2 with AD surrogate method leads to slightly better hypervolume and generational distance than the baseline in the nominal case.



(a) Hypervolume

(b) Generational distance

Figure 5.22 – High level metrics for all the approaches using the RECO scheduler and complex infrastructure.

5.7 Discussion

Many experiments are presented along the previous sections, showing different aspects of the proposed approach. For sake of clarity, the main findings are sorted in different categories. Some points concerning SPEA2 and our modified version USPEA2 are stated first. Then, the important observations on the proposed function approximation methods are described, followed by the identified limits of such approach.

5.7.1 USPEA2

The proposed modifications of SPEA2 to design USPEA2 aim at improving the performance when the individuals consist in a mix of evaluated and approximated solutions (through online-trained surrogate model). However, these modifications could affect the general behavior of this multi-objective evolutionary algorithm.

No significant difference is observed compared to SPEA2 when no surrogate method is used, as shown by the results in section 5.6.2. As expected however, when an approximation function is used the results are usually improved compared to SPEA2. It is clearly shown for the hypervolume of MLP and RBF surrogate methods. When a relatively high budget of costly evaluations is used ($eval_{max} \geq 400$), using USPEA2 also allows to keep increasing hypervolume for AD and MHT. This is explained by the *reliable archive* used, preventing the loss of good solutions when surrogate models are too much optimistic, giving falsely believed, unreliable good solutions which replace the reliable ones. The additional results in other tested conditions (scenarios and number of time steps), while not detailed here because of the lack of any other interest, also give similar results.

While it was expected to be similar, the impact of USPEA2 on the average distance to the Pareto front (GD) is lower and in some situations may even leads to worst (higher) values. This phenomenon is caused mainly by the set of solutions lost, which are observed contain more often some of the extremum of the current front. It causes a diminished spreading, making easier to keep only a few solutions close to a given area of the true Pareto front. In addition, because the generational distance only takes non-dominated solutions into account, any individual resulting in very poor outcomes is not impacting this metric.

5.7.2 Online surrogate of objective functions

In addition to the method from the literature, multilayer percetrons (MLP) and radial-basis functions (RBF), two methods to new surrogate methods for time series, high-dimensonality objective functions are presented. The average-distance based method (AD) is a trivial approach and appear to perform relatively badly when used alone, as detailed in section 5.5.2. On the other hand, the multiresolution Haar wavelet distance method (MHT) is designed to handle timeseries better and provides quite accurate prediction according the results in section 5.5.2. The two commonly used surrogate methods exhibit bad results when evaluated alone and with the reduced training set caused by the limited budget of costly evaluation in our problem. When used together with an evolutionary algorithm however, the surrogate method used could impact the solution explored, which in turn impacts the known solutions used to build the surrogate and therefore the quality of subsequent approximations.

The different Pareto-based characteristics of the surrogate methods, described in section 5.5.2, are expected to impact the quality of the overall optimization. Most experiments using MHT approximation give a better hypervolume. Notably, the results of fig. 5.17a show it reaches similar or higher hypervolume compared to SPEA2 alone with a budget 4 times less important. However it does not reduce significantly the generational distance in the experiments performed on the simplified model. This is also confirmed by the experiments made with the *Best* initial scheme, shown in fig. 5.18a, where MHT approximation appear to improve only slightly the hypervolume when the individuals are already well spread. Such results suggest that using MHT prevents costly evaluation of many individuals near to the known Pareto-optimal solutions, allowing a faster spreading, while failing to identify accurately non-dominated individuals among them (which are closer to the real Pareto front from only a small difference).

Compared to MHT, the behavior of the overall optimization is quite different when AD is used. While it leads to few or no improvement of hypervolume in most cases, it results in a slightly better *GD* for large budget (≥ 400) of costly evaluations. It also performs better than all the other methods when the individuals are already well spread, as shown for the *Best* initial population scheme. This is likely explained by a low prevention of individuals evaluation near the current approximation of the Pareto front. As suggested by preliminary results in section 5.5.2, it is caused by both high approximation miss rate and *pessimistic* estimations (almost the totality of dominated individuals are considered such).

Both RBF and MLP only improve the quality of the optimization in a limited amount of cases. The multilayer percetrons surrogate gives good results compared to the baseline in a specific scenario (*FewRenew*) and for large solution space (time window with 320 time steps). Oppositely, using a surrogate based on radial-basis functions performs better with small solution space. RBF also gives very good results for both hypervolume and generational distance for the largest budget tested, by knowing 1600 solutions at the end. The two points confirms that RBF is remarkably well-suited when the ratio of the size of the training set over the number of dimensions of the problem is higher than our target.

The most important result is the *stability* of the proposed MHT method over the various tested conditions. First, it is less affected than the other methods by the formulation of the objective functions (scenarios and internal values such as ghg_{sell} used to compute the PDM utility). It is also the only method which gives significant improvements when a more complex IT infrastructure is considered, using an arguably realistic scheduler from the literature in place of the simplified model. Then it grants a significant improvement of hypervolume for any target budget of costly evaluations. Finally it is clearly robust

to the high-dimensionalty aspect of large time series, resulting in relative hypervolume improvement similar or greater for values of T from 20 to 320. This new method may seem simpler than approaches such as RBF and MLP. It is however designed deeply for taking the time series aspect into account, mitigating the issues of high-dimensional space, with the results confirming this initial hypothesis.

5.7.3 Limits of approximation-based methods

When using a surrogate model to approximate black-box functions, evaluation of the quality of the approximations is required in order to choose the best model and to tune its parameters. In this work, several difficulties are exhibited. Working with multiple objectives makes it more delicate to define what is the *quality* or *accuracy* of a set of surrogate models. As highlighted by Karakasis and Giannakoglou [98], if a single objective is badly approximated, a good solution appears as dominated and therefore uninteresting. Some metrics, designed for multi-objective approximation, are proposed in section 5.5.2. Unfortunately, maximizing all these metrics is by itself a multi-objective problem and it is likely that a trade-off must be carefully selected depending on the target problem.

A second issue is related to the interactions between the meta-heuristic (SPEA2 or USPEA2) and the online surrogate model itself, as explained in section 5.5. The overall algorithm quality becomes difficult to predict, as it does not depend solely on the accuracy of the approximation functions used. A first illustration is given in section 5.5.2, with the quality metrics changing significantly for some surrogate methods when measured during the algorithm execution. The difference of behaviors of each approximation method depending only on the initial population used is also an issue which is not fully covered in this study.

There is clearly an advantage, according to the different experiments, in combining surrogate model to meta-heuristics when objective functions are costly. These issues, however, make very difficult to predict of the *actual improvement* before actually running the algorithm to evaluate the overall results.

5.8 Conclusion

In this chapter, we presented an approach for multi-objective power planning optimization between a datacenter and an electrical infrastructure with on-site RES, each having its own objective and being considered as a black-box. Multi-objective heuristics being especially costly, a surrogate-assisted method was explored in order to reduce the amount of evaluation of solutions required to find a set of good power planning trade-offs. Two surrogate methods are proposed, Average Distance (AD) and Multiresolution Haar Transform (MHT), designed to deal efficiently with time series. In addition, a variant of the multi-objective evolutionary algorithm SPEA2, named USPEA2 is presented to handle better a mix of individuals with evaluated objectives and unreliable, approximated values.

Extensive experiments have been presented using a simplified datacenter and electrical sources model, both to study the impact of the parameters of the algorithms and to evaluate the performance in various conditions. By improving the quality and the stability of the results when a surrogate method is used, USPEA2 proved to be a robust MOEA for leveraging online objective function approximation. The proposed MHT surrogate method led to a clear improvement of the quality of the solution set when used together with USPEA2 in all tested conditions compared to other methods and absence original SPEA2 algorithm without surrogate. By taking the time series aspect of the solution

space, this method gives better results than techniques known to provide good quality surrogates in the literature, such as multilayer neural networks and radial-basis functions. The evaluation using a more complex and arguably rich datacenter and workload model, with a scheduler from the literature confirms the benefit of the proposed approach in real use cases.

To the best of our knowledge, this is the first work on surrogate-based evolutionary optimization targeting objective functions defined on time series. Hence it opens several perspectives which were not explored here, such as the usage of crossover and mutation functions designed for this particular case of high-dimensional space. Other improvements may be obtained by using techniques used in similar areas. The good results obtained by the radial-basis functions surrogate when the evaluation budget is high suggests it may be a good candidate for hybrid surrogate method together with the MHT method proposed.

The work presented here is also the first to attempt a multi-objective optimization of datacenter and rich electrical infrastructure while being totally agnostic of the respective models. Such a black-boxes model has tangible benefits in real world situations, by reducing the coupling between electrical and IT aspects, which are in practice managed by separated entities and evolving independently. The multi-objective aspect, in a Pareto sense, avoids to decide the articulation of objectives beforehand and allows to use arbitrary objectives for the IT and the electrical parts. In this work, we only focused on 2 objectives, one for each part of the infrastructure. However the approach itself supports more objectives. A possibly interesting extension is the addition of an objective common to both parts such as the economical aspects.

By tackling the multi-objective aspect without prior assumption on the articulation of objectives, the choice of a solution among the best trade-offs (Pareto-optimal solutions) has not been studied. Without deeper investigation, this approximation of the Pareto-front can be used to implement simple policies or to improve the choice of an existing decision maker. For instance, the *knee points* of the Pareto-front give the most interesting trade-offs [20, 194], which is helpful to articulate the preferences at a given time. By providing a set of good trade-offs, an automatic decision making algorithm can take into account both *a priori* articulation of objectives and *a posteriori* knowledge (shape of the Pareto-front, knee points, extremum solutions). Even with the improvements proposed here, finding the Pareto-front with costly ITDM and PDM is quite time consuming. Hence a possible use case is in a two-phases optimization process such as proposed by Pahlevan et al. [148], in which it can take a good offline decision for a middle-term horizon and is completed by cheap online heuristic to adjust the decision to small deviations from the power planning.

When a longer period than the time window is considered, the problem can also be modeled as multi-objective sequential decision making problem: at each decision epoch, a solution (power planning for the upcoming time window) must be chosen among a set of Pareto-optimal trade-offs. In such formulation, a multi-criteria reward is obtained depending on the chosen solution, represented here by the ITDM and PDM utilities and their internal state is affected by the decision, such as the ESD state of charge or the some waiting workload. Hence, the best long-term decision for maximizing the accumulated reward depends of its impact on the overall state in addition to the immediate reward. In this chapter, the black-box model is used to hide these states, making hard to reason on a time longer than a decision epoch. Future studies could explore how to tackle this long-term decision making problem with different amount of information on the ITDM and PDM states.

Another perspective explored by Kamjoo et al. [97] in a slightly different context

(RES powering a fixed target load) is to incorporate the *uncertainty* of the solutions as an independent objective. Along this chapter, none of the IT or electrical models considered any sources of uncertainty. But once a decision is implemented, the prediction of renewable energy production for instance is likely to not match exactly the real production. Hence we can imagine that some of the power profiles considered as potential solutions are more likely than others to result in undesirable situation (empty storage devices or massive lack of renewable energy compared to the prediction). With the presented approach, the miss-predictions could be addressed by two complementary ways: being used in a two-phases optimization as mentioned earlier and by running the proposed multi-objective optimization when the deviation compared to the previous power planning became too important. These two mechanisms are external to the approach itself, as it does not handle uncertain models. In the next chapter, these uncertainty are instead considered as a central dimension of the problem.

Chapter 6

Datacenter management with uncertain renewable energy production

Come, let us hasten to a higher plane,
Where dyads tread the fairy fields of Venn,
Their indices bedecked from one to n,
Commingled in an endless Markov chain!
...
I'll grant thee random access to my heart,
Thou'lt tell me all the constants of thy love;
And so we two shall all love's lemmas prove,
And in bound partition never part.

— Stanisław Lem, *The Cyberiad*

Contents

In the two previous chapters, we focused on the intermittent aspect of renewable energy sources (RES), proposing methods to take the varying availability of the energy into account in the management of a datacenter. These contributions, similarly to most of the literature, overlooked the uncertainty of solar and wind energies production forecasting.

We propose here to study a model with uncertain RES production. As detailed in section 2.4.2, most of the current works on datacenter management with RES are either ignoring the production uncertainties, or using simple methods to mitigate their effects, such as an online adjustment of the initial optimization. In order to strengthen the knowledge and provide a robust base for future studies, this problem is tackled here in a more theoretical way, assuming a full knowledge of RES production uncertainties (*i.e.* their probability distributions).

Contrary to the previous contributions, the infrastructure studied is a stand-alone microgrid, not connected to an electrical grid. A renewable source, photovoltaic (PV)

panels, is the only primary source and batteries are used to store some of its production. The datacenter is therefore constrained by the energy production and must carefully plan its power consumption as no grid backup can be used.

In this chapter, we propose a Markov Decision Process (MDP) formulation that tackles several flavors of the problem. We increase gradually the complexity of the problem, so that we analyze the structural properties of every instance and expose the impact of each additional constraint on the problem. Doing so gives a precious, deep understanding of the overall problem.

Markov decision processes, as explained in section 2.1.2, are generally impossible to solve exactly with continuous states and actions which are required in some of the models studied here. Generic discretization techniques can be applied to approximate the solution, but their computation is costly when a very close approximation is desired. The analysis conducted along this chapter reveals some structural properties of the optimal policy for workload management with uncertain renewable production. These properties can be leveraged in order to solve more efficiently the problem compared to a generic solution. To illustrate how these findings are usable in practice, a method for computing a bounded-error approximation of the optimal policy is described and implemented. The improvements provided by this uncertainty aware approach are attested by comparing it to the optimal policy with deterministic RES forecast and to the best achievable results using a perfect oracle. Its performances in terms of execution time to solve different instances of the problem are also attested experimentally.

The general infrastructure model studied and the overall problem are first described in section 6.1. Then the analysis of a first variant of the problem, considering perfect batteries and discrete state and action spaces is conducted along section 6.2. Section 6.3 extend the analysis to a second, more complex model, taking into account the efficiency of non-ideal batteries. The optimal policy structure described for this second variant is used in section 6.4 to describe the bounded approximation algorithm, which is implemented and studied experimentally. Section 6.5 concludes this chapter and present the main perspectives of this work.

6.1 Framework

6.1.1 Context

We consider a datacenter composed of machines that can execute a workload and of batteries that can store and return energy. The datacenter is not connected to the grid but to sources of renewable energy, so that the machines can be supplied by the batteries and/or directly by these renewable energy sources. Through misuse of language we will call *renewable energy* the energy that is produced by renewable energy sources.

A finite time window (or horizon) is considered, with discrete time model. At the beginning of each time step, users submit to the platform some workload to be executed during the time step, modeled as an homogeneous amount of work (*i.e.* millions of instructions). The platform manager decides at this point to commit to execute all or any part of the workload, qualified as *accepted* work thereafter. The power consumption is modeled as an energy proportional datacenter, which is a major simplification but becoming closer to the realty with the recent progress in this direction [91, 191].

The future renewable energy production is not known surely, but a forecast of energy production from renewable sources is available at the beginning of the time step. Therefore, the decision of the quantity of accepted work takes place before knowing the exact energy

production for the next time step. Each unit of work that has been accepted and that is actually executed during the time step provides a gain to the datacenter, while the datacenter gives back a penalty to the users for any work that has been accepted but could not be run. Submitted but non-accepted work does not imply any gain or loss to the datacenter.

Batteries are automatically charged if renewable energy is in excess, or discharged (down to 0) if not enough renewable energy is available to fulfill the datacenter needs, since there is no advantage in not executing all the accepted work to save energy in the batteries in this model. We consider that when there is not enough energy to supply the running machines during the time period, the energy is "gathered" to supply only a subset of the machines for the whole time period: the datacenter receives then a gain (resp. gives back a penalty) for any work that has been executed (resp. accepted but not executed).

6.1.2 Models

For the sake of understanding, we begin with a simple model of batteries and workload, and move step by step towards a more realistic setting. Two main variants are studied. In a first part, we consider ideal batteries, *i.e.* such that no battery loss takes place when charged or discharged, denoted as IDEAL variant. In a second part (IMPERFECT), we deal with non-ideal batteries, that suffer from imperfect efficiency, leading to energy losses when discharged and charged.

For each of these variants, three submodels are progressively considered. In the IDEAL-UNBOUNDED model, the batteries do not have maximum capacity and the available work at any step is infinite (the datacenter does not have other limitation on the total work that it can accept and execute during a time step than the available energy). In the IDEAL-BATTERYBOUNDED model, we are given a capacity on the batteries, *i.e.* the total stored energy is limited at any time. Finally, in the IDEAL-BOUNDED model, the users submit a finite amount of work, that is known to the datacenter before deciding which part will be accepted. Similarly, the non-ideal batteries case is studied across these three flavors, namely IMPERFECT-UNBOUNDED, IMPERFECT-BATTERYBOUNDED and IMPERFECT-BOUNDED.

We formalize this problem through Markovian Decision Processes (MDP), briefly presented in section 2.1.2 and start with some notations.

6.1.3 Formalization

We consider a finite-horizon MDP decomposed into $N-1$ periods, numbered from 1 to $N-1$, where period t starts at decision epoch t and finishes at decision epoch $t+1$.

A state of the MDP is fully characterized with the amount of energy stored in the batteries: we denote by $b_t \in \mathcal{B}$ the amount of energy stored in the battery at decision epoch t.

$(w_t)_{t\in\{1,\dots,N-1\}}$ is a vector that expresses the amount of work that is available at decision epoch t, *i.e.* the quantity of work that has been submitted by the users to the datacenter to be executed at period t. We denote by E_t the amount of renewable energy that is produced during period t. Its realization is denoted by $e_t \in \mathcal{E}$. Forecast informations give the probabilistic distribution of each E_t and its realization is known after decision epoch t.

We denote by $\mathcal{A}$ the default set of available actions. At decision epoch t, $a_t \in \mathcal{A}_t = \mathcal{A} \cap [0, w_t]$ is the amount of work that is accepted by the datacenter operator. D_t (resp. S_t) is the amount of work that is done (resp. spoiled, *i.e.* accepted but not run) by the platform during period t. These quantities are such that $D_t + S_t = a_t$. We define

$c^+ \in \mathbb{R}_\star^+$ as the gain earned by the datacenter per unit of work that has been accepted and executed, while $c^- \in \mathbb{R}_\star^+$ is the penalty that has to be paid back per unit of work that has been accepted but not executed during the time period. The revenue R_t of the datacenter during period t is then given by: $R_t = c^+ \cdot D_t - c^- \cdot S_t$.

We denote by $r_t(b_t, a_t)$ the expectation of the reward during period t, when, at decision epoch t, b_t units of energy are stored in the batteries and the datacenter operator chooses to accept a_t units of work. No decision is made at decision epoch N, and the final reward is there $r_N(b) = 0$ for all b (no bonus for the energy stored in the batteries).

The decision problem is to find possibly optimal policies, *i.e.* a sequence $(\pi_t)_{t\in\{1,\dots,N-1\}}$, where for all $t \in \{1, \dots, N-1\}$, $\pi_t : \mathcal{B} \to \mathcal{A}$ gives the action a_t to perform depending on the state b_t, such that the expectation of the sum of the rewards $\sum_{t=1}^{N-1} R_t$ is maximized.

For convenience, we define the optimal value functions V_t such that $V_t(b_t)$ is the maximum reward expectation that can be obtained when b_t units of energy are in the batteries at decision epoch t. It should be noticed that for all $b \in \mathcal{B}$, $V_N(b) = r_N(b) = 0$. We also denote by $Q_t(b_t, a_t)$ the expected reward in the following situation: choosing action a_t when in state b_t at decision epoch t, then an optimal sequence of actions until the end. Therefore, we have for all $b_t \in \mathcal{B}$, $V_t(b_t) = \max_{a\in\mathcal{A}_t} Q_t(b_t, a)$. Any action $a_t^\star \in \arg\max_{a\in\mathcal{A}_t} Q_t(b_t, a)$ will be called indifferently a maximizer or an optimal action at decision epoch t for a given b_t.

This generic formalization comes into several flavors according to the targeted model. If the batteries are ideal, we can use discrete random variables and states. Especially, $\mathcal{E} = \mathbb{N}$ and a compact notation q_t^e is used for the probability that e units of energy will be produced during period t, equivalently to $P[E_t = e]$. As the energy model of the datacenter is proportional to the load, we consider, without loss of generality, the units of work being equivalent to the corresponding energy consumption. Similarly, the time unit corresponding to the duration of a time step, we can mix the energy, workload and power units without the burden of constant factors added in all the equations. Hence, the charge and discharge of the batteries are ruled by: $b_{t+1} = b_t + e_t - d_t$, where e_t (resp. d_t) is the realization of E_t (resp. D_t).

The Ideal-Unbounded model corresponds to $\mathcal{B} = \mathbb{N}$, and for all $t \in \{1, \dots, N-1\}$, $\mathcal{A} = \mathbb{N}$ and $w_t = \infty$. In the Ideal-BatteryBounded model, we denote by $B^{\max}$ the capacity of the batteries, *i.e.* $\mathcal{B} = \{0, \dots, B^{\max}\}$, while $\mathcal{A}$ and the w_t are unchanged. In the Ideal-Bounded, $\mathcal{B}$ is unchanged, and for all $t \in \{1, \dots, N-1\}$, $w_t \in \mathbb{N}$ and the set of available actions at decision epoch t is $\mathcal{A}_t = \mathcal{A} \cap [0, w_t]$.

In the case of non-ideal batteries, there is an energy loss when the batteries are charged or discharged. Without loss of generality however, we can assume that this lack of efficiency is paid only when charged, *i.e.* there exists $\eta \leq 1$ such that if the batteries are charged with e units of energy, only ηe units of energy are stocked into the batteries. The charge and discharge of the batteries follows:

$$b_{t+1} = \begin{cases} b_t - (d_t - e_t) & \text{if } d_t \geq e_t \\ b_t + \eta \cdot (e_t - d_t) & \text{otherwise} \end{cases}$$

This results in a slightly modified model: the available work, the accepted work, the renewable energy production and the remaining energy in the batteries are not discrete but continuous[1]. Especially, $\mathcal{E} = \mathbb{R}^+$. We denote by q_t the probability density function of E_t, and by F_t its cumulative distribution. In addition, we need some regularity in

[1]For sake of readibility, and according to the ISO 31-11 standard on *mathematical signs and symbols for use in physical sciences and technology*, we use the form $]x, y[$ to refer to an open interval between x and y, instead of the more common (x, y).

the probabililty density function: we assume that for all $t \in \{1, \dots, N\}$, q_t is in $\mathcal{C}^1$. We emphasize that the production of renewable energy is intrinsically bounded by physical limitations, *e.g.* the size of the solar panels, hence there exists $E^{\max}$ such that for all $t \in \{1, \dots, N\}$ and $e \in [E^{\max}, \infty[$, $q_t(e) = 0$.

In the Imperfect-Unbounded model, $\mathcal{B} = \mathbb{R}^+$, and for all $t \in \{1, \dots, N-1\}$, $\mathcal{A} = \mathbb{R}^+$ and $w_t = \infty$. In the Imperfect-BatteryBounded model, we denote by $B^{\max}$ the capacity of the batteries, *i.e.* $\mathcal{B} = [0, B^{\max}]$, while $\mathcal{A}$ and the w_t are unchanged. In the Imperfect-Bounded, $\mathcal{B}$ is unchanged, and $w_t \in \mathbb{R}^+$ and the set of available actions at decision epoch t is $\mathcal{A}_t = \mathcal{A} \cap [0, w_t]$.

6.2 Ideal batteries

6.2.1 Unbounded batteries and workload

We consider first the Ideal-Unbounded model where neither the batteries nor the workload that can be executed on the platform during a period are bounded: $\mathcal{B} = \mathbb{N}$ and for all t, $w_t = \infty$.

We define $p_t(b'|b, a)$ as the probability that there is b' units of energy in the batteries at decision epoch $t+1$ when there was b units of energy in the batteries at decision epoch t and the datacenter operator accepted a units of work.

By definition,

$$Q_t(b, a) = r_t(b, a) + \sum_{b' \in \mathcal{B}} p_t(b'|b, a) \cdot V_{t+1}(b') \tag{6.1}$$

The reward function $r_t(b, a)$, which is the expectation of the reward that is received during period t when we choose action a while being in state b at decision epoch t, can be decomposed into two cases. If $a \leq b$, there is enough energy in the batteries not to depend on the renewable energy, so that we are certain that the accepted work is executed, hence $r_t(b, a) = a \cdot c^+$. If $a > b$, we rely on the renewable energy, whose value is e with probability q_t^e. If $e \geq a - b$, the sum $b + e$ of available energy is not less than a, thus the reward is $a \cdot c^+$. If $e < a - b$, then we only execute $b + e$ units of work out of the a accepted units of work: we earn $(b + e) \cdot c^+$ but we pay a penalty of $(a - (b + e)) \cdot c^-$ for the accepted but not run work. More formally,

$$r_t(b, a) = \begin{cases} \sum_{e=0}^{a-b-1} q_t^e \cdot (c^+ \cdot (b + e) - c^- \cdot (a - b - e)) \\ \quad + \sum_{a-b}^{\infty} q_t^e \cdot c^+ \cdot a & \text{if } a > b \\ c^+ \cdot a & \text{otherwise,} \end{cases}$$

which simplifies as

$$r_t(b, a) = \begin{cases} c^+ \cdot a - (c^+ + c^-) \sum_{e=0}^{a-b-1} q_t^e \cdot (a - b - e) & \text{if } a > b \\ c^+ \cdot a & \text{otherwise} \end{cases} \tag{6.2}$$

We will proceed by backward induction, starting at the last decision epoch $N - 1$, storing the best outcomes in V_{N-1} for the previous step. Since there is no bonus for the energy that could be stored in the batteries at the final decision epoch (*i.e.* $V_N = 0$), the value iteration V_{N-1} at decision epoch $N - 1$ is fully determined by the reward that is obtained during period $N - 1$.

Lemma 1. *In the* Ideal-Unbounded *model, for all* $b \in \mathcal{B}$,

$$V_{N-1}(b) = c^+ \cdot (\alpha_{N-1} + b) - (c^+ + c^-) \sum_{e=0}^{\alpha_{N-1}-1} q_{N-1}^e \cdot (\alpha_{N-1} - e), \tag{6.3}$$

where α_{N-1} is the unique integer that fulfills

$$P\left[E_{N-1} \leq \alpha_{N-1}\right] \leq \rho < P\left[E_{N-1} \leq \alpha_{N-1}+1\right], \quad \text{where } \rho = \frac{c^+}{c^+ + c^-}. \tag{6.4}$$

Choosing action $a^\star_{N-1} = \alpha_{N-1} + b$ at the last decision epoch $N-1$ is optimal with a battery b.

Proof. Let $b \in \mathcal{B}$; we try to find some a that maximizes $r_{N-1}(b,a)$ by studying how $r_{N-1}(b,a)$ evolves with a. Based on the definition of the reward function from eq. (6.2), the following holds

$$\begin{aligned} r_{N-1}(b,a+1) - r_{N-1}(b,a) &= c^+ - (c^+ + c^-) \cdot \sum_{e=0}^{a-b} q^e_{N-1} \\ &= c^+ - (c^+ + c^-) \cdot P\left[E_{N-1} \leq a-b\right] \\ r_{N-1}(b,a+1) - r_{N-1}(b,a) &= (c^+ + c^-) \cdot (\rho - P\left[E_{N-1} \leq a-b\right]), \end{aligned}$$

where $\rho = \frac{c^+}{c^+ + c^-}$. As a function of $\delta \in \mathbb{Z}$, $P\left[E_{N-1} \leq \delta\right]$ is non-decreasing, $P\left[E_{N-1} \leq -1\right] = 0$ and $P\left[E_{N-1} \leq \delta\right] \xrightarrow{\delta \to \infty} 1$, and $\rho \in]0,1[$; therefore there exists a unique $\alpha_{N-1} \in \mathbb{N}$ such that:

$$P\left[E_{N-1} \leq \alpha_{N-1}\right] \leq \rho < P\left[E_{N-1} \leq \alpha_{N-1}+1\right].$$

$r_{N-1}(b,a)$ is a non-decreasing function of a on $\{0,\dots,\alpha_{N-1}+b\}$, and non-increasing on $[\alpha_{N-1}+b, \infty[\cap \mathbb{N}$, hence the theorem. □

For each of the previous decision epochs, there are several optimal actions that maximizes the sum of the expected rewards until the end. They are described in the following lemma.

Lemma 2. *In the* Ideal-Unbounded *model, for all $t \in \{1,\dots,N-2\}$, for all $b \in \mathcal{B}$,*

$$\begin{cases} V_t(b) = V_t(0) + c^+ \cdot b \\ V_t(0) = V_{t+1}(0) + c^+ \cdot E\left[E_t\right] \end{cases} \tag{6.5}$$

A policy that chooses, for all $t \in \{1,\dots,N-2\}$, $a^\star_t \in \{0,\dots,b\}$, then $a^\star_{N-1}$ at decision epoch $N-1$ is optimal.

Proof. We proceed in two steps, by induction. Let $(\mathcal{H}_t)$ be the following induction hypothesis: for a given $t \geq 1$, for all $b \in \mathcal{B}$,

$$V_t(b) = V_t(0) + c^+ \cdot b.$$

We assume first that there exists $t > 1$ such that $(\mathcal{H}_{t+1})$ is verified, and prove that $(\mathcal{H}_t)$ is then true.

We recall that

$$Q_t(b,a) = r_t(b,a) + \sum_{b' \in \mathcal{B}} p_t(b'|b,a) \cdot V_{t+1}(b').$$

In order to express the transition probabilities $p_t(b'|b,a)$, we rely on the same line of reasoning that we used to express $r_t(b,a)$ in eq. (6.2). If the renewable energy e is such that $b+e \geq a$, then a units of energy are consumed to run the accepted work and there remains $b+e-a$ energy in the batteries at the beginning of the next decision epoch. Otherwise, the

whole available energy is consumed (and does not reach a), and the batteries are empty at the end of the period. In other words,

$$p_t(b'|b,a) = \begin{cases} P\left[E_t = e\right] & \text{if } b' = e + b - a \\ P\left[E_t < a - b\right] & \text{if } b' = 0 \\ 0 & \text{otherwise,} \end{cases}$$

which can be rewritten as

$$p_t(b'|b,a) = \begin{cases} q_t^e & \text{if } b' = e + b - a \\ \sum_{e=0}^{a-b-1} q_t^e & \text{if } b' = 0 \wedge a - b > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Copying the reward expression of eq. (6.2) in the expression of $Q_t(b,a)$, we obtain

$$Q_t(b,a) = \begin{cases} \sum_{e=0}^{a-b-1} q_t^e \cdot \Big(V_{t+1}(0) + c^+ \cdot (b+e) - c^- \cdot (a-b-e)\Big) \\ \quad + \sum_{a-b}^{\infty} q_t^e \cdot \Big(V_{t+1}(b+e-a) + c^+ \cdot a\Big) & \text{if } a > b \\ \sum_{e=0}^{\infty} q_t^e \cdot V_{t+1}(b+e-a) + c^+ \cdot a & \text{otherwise.} \end{cases}$$

By induction hypothesis,

$$Q_t(b,a) = \begin{cases} \sum_{e=0}^{a-b-1} q_t^e \cdot \Big(V_{t+1}(0) + c^+ \cdot (b+e) - c^- \cdot (a-b-e)\Big) \\ \quad + \sum_{a-b}^{\infty} q_t^e \cdot \Big(c^+ \cdot (b+e-a) + V_{t+1}(0) + c^+ \cdot a\Big) & \text{if } a > b \\ \sum_{e=0}^{\infty} q_t^e \cdot \Big(c^+ \cdot (b+e-a) + V_{t+1}(0)\Big) + c^+ \cdot a & \text{otherwise.} \end{cases}$$

Since $(q_t^e)_{e\in\mathbb{N}}$ represents a probability distribution on $\mathbb{N}$, $\sum_{e=0}^{\infty} q_t^e = 1$, hence

$$Q_t(b,a) = \begin{cases} V_{t+1}(0) + c^+ \cdot \sum_{e=0}^{\infty} q_t^e \cdot (b+e) - c^- \cdot \sum_{e=0}^{a-b-1} (a-b-e) & \text{if } a > b \\ V_{t+1}(0) + c^+ \cdot b + c^+ \cdot E\left[E_t\right] & \text{otherwise} \end{cases}$$

$$Q_t(b,a) = \begin{cases} V_{t+1}(0) + c^+ \cdot b + c^+ \cdot E\left[E_t\right] - c^- \cdot \frac{1}{2} \cdot (a-b+1) \cdot (a-b) & \text{if } a > b \\ V_{t+1}(0) + c^+ \cdot b + c^+ \cdot E\left[E_t\right] & \text{otherwise.} \end{cases}$$

We can deduce that for a given b, Q_t is constant in a when $a \leq b$ and is not greater than this constant when $a > b$. We have indeed

$$\forall (b,a,a') \in \mathcal{B} \times \mathcal{A}^2\ ,\ (a \leq b) \wedge (a' \geq b) \Rightarrow Q_t(b,a') \leq Q_t(b,a),\ \text{and}$$

$$\forall (b,a,a') \in \mathcal{B} \times \mathcal{A}^2\ ,\ (a \leq b) \wedge (a' \leq b) \Rightarrow Q_t(b,a') = Q_t(b,a).$$

For all $b \in \mathcal{B}$, $V_t(b)$ is defined as $V_t(b) = \max_{a\in\mathcal{A}} Q_t(b,a)$. According to the two previous equations, the maximum is attained for any $a \leq b$. Especially, $V_t(0) = V_{t+1}(0) + c^+ \cdot E\left[E_t\right]$, which implies that $(\mathcal{H}_t)$ is true.

Instantiating eq. (6.3) of lemma 1 with $b = 0$ shows that $(\mathcal{H}_{N-1})$ is true, hence $(\mathcal{H}_t)$ is true for any $t \in \{1, \dots, N-1\}$ by induction.

The previous proof also shows that for any $t \in \{1, \dots, N-2\}$,

$$V_t(0) = \max_{a \in \mathcal{A}} Q_t(0, a) = Q_t(0, 0) = V_{t+1}(0) + c^+ \cdot E\left[E_t\right],$$

which demonstrates the lemma.

□

Remark 1. *Lemma 2 shows that in this simple problem there exist several optimal policies. One of them would be not to accept any work, and store the energy until the last moment, then, at decision epoch $N-1$, choosing the right action with the help of lemma 1. This trivial policy will not be optimal any more in the next* Ideal-BatteryBounded*, since energy could be spoiled when the batteries become full.*

6.2.2 Bounded Battery and Unbounded Workload

We take one step towards a more realistic model, through Ideal-BatteryBounded, where we set a maximum capacity to the battery: there exists $B^{\max} \in \mathbb{N}$ such that $\mathcal{B} = \{0, \dots, B^{\max}\}$.

In this model, the reward function is the same as in Ideal-Unbounded model, given in eq. (6.2) while the state transition function is slightly modified. We transition to state $b' = B^{\max}$, when the renewable energy is not less than $B^{\max} - b + a$ (the excess energy is spoiled):

$$p_t(b'|b, a) = \begin{cases} P\left[E_t = e\right] & \text{if } b' = e + b - a \wedge b' \in \{1, \dots, B^{\max} - 1\} \\ P\left[E_t < a - b\right] & \text{if } b' = 0 \\ P\left[E_t > B^{\max} - b + a\right] & \text{if } b' = B^{\max} \\ 0 & \text{otherwise.} \end{cases} \tag{6.6}$$

Contrary to the Ideal-Unbounded model, we cannot here exhibit a closed formula for $V_t(b)$. However, we show in lemma 3 an important property stipulating that $Q_t(b+1, a+1) = Q_t(b, a) + c^+$, which means that accepting one more unit of work while having one more unit of energy in the batteries leads to an additional reward of c^+. This implies that the behavior of $Q_t(b, a)$ when $a \geq b$ can be inferred from the behavior of $Q_t(0, a)$. We prepare the final induction proof by demonstrating in lemma 4 a property that will be useful to show that the optimal action cannot be less than b. We focus then on $a \mapsto Q_t(0, a)$, deriving properties (that leads to the existence of a maximum) in lemma 5. Finally, we compile the results in theorem 1, expressing $V_t(b)$ according to $V_t(0)$ and ensuring that the properties on the value iteration are preserved by induction.

Lemma 3. *For all $b \in \{0, \dots, B^{max} - 1\}$ and $a \in \mathcal{A}$,*

$$Q_t(b+1, a+1) = Q_t(b, a) + c^+. \tag{6.7}$$

Proof. First, from the definition of p_t in eq. (6.6) we observe that $p_t(b'|b+1, a+1) = p_t(b'|b, a)$ since b and a appear only in the form of $a - b$.

Similarly, based on eq. (6.2),

$$r_t(b+1,a+1) = \begin{cases} c^+ \cdot (a+1) \\ \quad -(c^+ + c^-) \sum_{e=0}^{a+1-b-2} q_t^e \cdot (a+1-b-1-e) & \text{if } a+1 > b+1 \\ c^+ \cdot (a+1) & \text{otherwise} \end{cases}$$

$$= c^+ + \begin{cases} c^+ \cdot a - (c^+ + c^-) \sum_{e=0}^{a-b-1} q_t^e \cdot (a-b-e) & \text{if } a > b \\ c^+ \cdot a & \text{otherwise} \end{cases}$$

$$r_t(b+1,a+1) = c^+ + r_t(b,a).$$

Using these properties in eq. (6.1), we show that

$$\begin{aligned} Q_t(b+1,a+1) &= r_t(b+1,a+1) + \sum_{b' \in \mathcal{B}} p_t(b'|b+1,a+1) V_{t+1}(b') \\ &= c^+ + r_t(b,a) + \sum_{b' \in \mathcal{B}} p_t(b'|b,a) V_{t+1}(b') \\ Q_t(b+1,a+1) &= c^+ + Q_t(b,a) \end{aligned}$$

□

Before going further, we can slightly rewrite eq. (6.6) to ease $Q_t(b,a)$ expression:

$$p_t(b'|b,a) = \begin{cases} q_t^e & \text{if } b' = e + b - a \wedge b' \in \{1, \dots, B^{\max}\} \\ \sum_{e=0}^{a-b} q_t^e & \text{if } b' = 0 \wedge a - b \geq 0 \\ \sum_{e=B^{\max}-b+a+1}^{\infty} q_t^e & \text{if } b' = B^{\max} \\ 0 & \text{otherwise.} \end{cases} \tag{6.8}$$

We first study in the following lemma 4 the behavior of Q_t for small actions, *i.e.* when the accepted work is not more than the energy stored in the batteries.

Lemma 4. *Given $t \in \{1, \dots, N-1\}$, if $V_{t+1}(b+1) - V_{t+1}(b) \leq c^+$ for all $b \in \{0, \dots, B^{max}-1\}$, then $Q_t(b,a)$ is non-decreasing in a in the interval $\{0, \dots, b\}$ for all $b \in \mathcal{B}$.*

Proof. Let $t \in \{1, \dots, N-2\}$, $b \in \mathcal{B}$ and $a \in \{0, \dots, b-1\}$.

Since the reward function is the same in both IDEAL-BATTERYBOUNDED and IDEAL-UNBOUNDED models, we have (*cf.* second case in eq. (6.2))

$$Q_t(b,a) = c^+ \cdot a + \sum_{e=0}^{B^{\max}-(b-a)} q_t^e \cdot V_{t+1}(b-a+e) + \sum_{e=B^{\max}-(b-a)+1}^{\infty} q_t^e \cdot V_{t+1}(B^{\max}), \text{ hence}$$

$$Q_t(b,a) = c^+ \cdot a + V_{t+1}(B^{\max}) \cdot E\left[E_t\right] + \sum_{e=0}^{B^{\max}+a-b} q_t^e \cdot (V_{t+1}(b-a+e) - V_{t+1}(B^{\max})).$$

We can then deduce that

$$Q_t(b,a+1) - Q_t(b,a) = c^+ + \sum_{e=0}^{B^{\max}+a-b} q_t^e \cdot (V_{t+1}(b-a-1+e) - V_{t+1}(b-a+e)). \tag{6.9}$$

As a consequence,

$$Q_t(b, a+1) - Q_t(b, a) \geq c^+ + \sum_{e=0}^{B^{\max}+a-b} q_t^e \cdot (-c^+)$$
$$Q_t(b, a+1) - Q_t(b, a) \geq 0,$$

since all q_t^e are positive and sum up to 1. This proves the lemma. □

The previous lemma will allow us discarding the small actions (*i.e.* $a \leq b$) when seeking for optimal actions. Lemma 3 tells that the behavior of $Q_t(b, a)$ for bigger actions is translated from $Q_t(0, a)$. That is why we focus on $Q_t(0, a)$ and show that after a given threshold, trying to accept more work keeps degrading the expected future total reward.

Lemma 5. *If for all $b \in \{0, \dots, B^{max} - 1\}$, $V_{t+1}(b+1) - V_{t+1}(b) \leq c^+ + c^-$ and if V_{t+1} is non-decreasing, then $Q_t(0, a)$ is non-increasing in a when $a \geq \alpha_t$, where α_t is the unique integer such that:*

$$P\left[E_t \leq \alpha_t\right] \leq \rho < P\left[E_t \leq \alpha_t + 1\right], \quad \text{where } \rho = \frac{c^+}{c^+ + c^-}.$$

Proof. Let $t \in \{1, \dots, N-2\}$ and $a \geq 0$. We use the formulae of the reward function (eq. (6.2), first case) and of the transition probailities in eq. (6.8) to compute $Q_t(0, a)$.

$$\begin{aligned} Q_t(0, a) = &\sum_{e=0}^{a} q_t^e \cdot \left(c^+ \cdot a - (c^- + c^+) \cdot (a - e) + V_{t+1}(0)\right) \\ &+ \sum_{e=a+1}^{B^{\max}+a} q_t^e \cdot \left(c^+ \cdot a + V_{t+1}(e - a)\right) + \sum_{e=B^{\max}+a+1}^{\infty} q_t^e \cdot \left(c^+ \cdot a + V_{t+1}(B^{\max})\right) \\ Q_t(0, a) = &\, c^+ \cdot a - (c^- + c^+) \sum_{e=0}^{a} q_t^e \cdot (a - e) \\ &+ \sum_{e=0}^{a} q_t^e \cdot V_{t+1}(0) + \sum_{e=a+1}^{B^{\max}+a} q_t^e \cdot V_{t+1}(e - a) + \sum_{e=B^{\max}+a+1}^{\infty} q_t^e \cdot V_{t+1}(B^{\max}) \end{aligned}$$

Let f and g such that

$$\begin{cases} f(a) = \sum_{e=0}^{a} q_t^e \cdot (a - e) \\ g(a) = \sum_{e=0}^{a} q_t^e \cdot V_{t+1}(0) + \sum_{e=a+1}^{B^{\max}+a} q_t^e \cdot V_{t+1}(e - a) + \sum_{e=B^{\max}+a+1}^{\infty} q_t^e \cdot V_{t+1}(B^{\max}) \end{cases}.$$

We then have $Q_t(0, a) = c^+ \cdot a - (c^- + c^+) \cdot f(a) + g(a)$.

For any function ϕ, we define $\Delta\phi(a)$ as $\Delta\phi(a) = \phi(a+1) - \phi(a)$ and $\Delta_a Q_t(0, a)$ as $\Delta_a Q_t(0, a) = Q_t(0, a+1) - Q_t(0, a)$. We show that Q_t is non-increasing in a by proving that $\Delta_a Q_t(0, a) \leq 0$. On the one hand,

$$\Delta f(a) = \sum_{e=0}^{a} q_t^e (a + 1 - e - (a - e)) + q_t^e (a + 1 - (a + 1))$$
$$\Delta f(a) = P\left[E_t \leq a\right].$$

On the other hand,

$$\begin{aligned}\Delta g(a) = q_t^{a+1} \cdot V_{t+1}(0) \\ + \sum_{e=a+2}^{B^{\max}+a+1} q_t^e \cdot V_{t+1}(e-a-1) - \sum_{e=a+1}^{B^{\max}+a} q_t^e \cdot V_{t+1}(e-a) \\ - q_t^{B^{\max}+a+1} \cdot V_{t+1}(B^{\max}).\end{aligned}$$

Then synchronizing the sums

$$\begin{aligned}\Delta g(a) = q_t^{a+1} \cdot V_{t+1}(0) - q_t^{B^{\max}+a+1} \cdot V_{t+1}(B^{\max}) \\ + \sum_{e=a+2}^{B^{\max}+a} q_t^e \cdot V_{t+1}(e-a-1) - \sum_{e=a+2}^{B^{\max}+a} q_t^e \cdot V_{t+1}(e-a) \\ + q_t^{B^{\max}+a+1} \cdot V_{t+1}(B^{\max}+a+1-a-1) - q_t^{a+1} \cdot V_{t+1}(1),\end{aligned}$$

which ends up with

$$\Delta g(a) = -\sum_{e=a+1}^{B^{\max}+a} q_t^e \cdot \Delta V_{t+1}(e-a-1).$$

Now, coming back to $Q_t(0,a)$,

$$\Delta_a Q_t(0,a) = c^+ - P\left[E_t \leq a\right] \cdot (c^+ + c^-) - \sum_{e=a+1}^{B^{\max}+a} q_t^e \cdot \Delta V_{t+1}(e-a-1) \tag{6.10}$$

To prove the lemma, we assume now that for all $b \in \{0, \dots, B^{\max}-1\}$, $V_{t+1}(b+1) - V_{t+1}(b) \leq c^+ + c^-$ and V_{t+1} is non-decreasing. This implies that all $\Delta V_{t+1}(\cdot)$ are non-positive, hence

$$\Delta_a Q_t(0,a) \leq c^+ - P\left[E_t \leq a\right] \cdot (c^+ + c^-).$$

We can deduce that as soon as $a \geq \alpha_t$, $P\left[E_t \leq a\right] \geq c^+/(c^+ + c^-)$, and $Q_t(0,a)$ is non-increasing in a, hence the lemma. □

Theorem 1. *In the* IDEAL-BATTERYBOUNDED *model, the value iteration (resp. the set of maximizers) at step t for all $b \in \{1, \dots, B^{max}\}$ are obtained from the value iteration (resp. the maximizers) in state 0 such as*

$$\forall t \in \{1, \dots, N-1\}, \begin{cases} V_t(b) = V_t(0) + b \cdot c^+ & \text{(6.11a)} \\ \pi_t^*(b) \supseteq \{a + b : a \in \pi_t^*(0)\}. & \text{(6.11b)} \end{cases}$$

Proof. Let $(\mathcal{H}_t)$ be the following induction hypothesis: $V_t(b+1) - V_t(b) \leq c^+$ and V_{t+1} is non-decreasing. We assume that for a given $t \in \{1, \dots, N-1\}$, $(\mathcal{H}_{t+1})$ is true and show that $(\mathcal{H}_t)$ is then true.

First, according to lemma 5, we know that $\{Q_t(0,a) : a \in \mathcal{A}\}$ has a maximum, and this maximum occurs when $a < \alpha_t$.

Next, according to lemma 3, we have for all $(b,a) \in \mathcal{B} \times \mathcal{A}$, $Q_t(b+1, a+1) = c^+ + Q_t(b,a)$, which implies by induction that $Q_t(b, a+b) = b \cdot c^+ + Q_t(0,a)$. Therefore

$$\{Q_t(b, a+b) : a \in \mathcal{A}\} = \left\{b \cdot c^+ + Q_t(0,a) : a \in \mathcal{A}\right\}$$

and $\{Q_t(b, a+b) : a \in \mathcal{A}\}$ has a maximum.

On the other side, $\{Q_t(b,a) : a \in \mathcal{A} \wedge a < b\}$ has a maximum and lemma 4 stipulates that $Q_t(b,a)$ is non-decreasing in a when $a \leq b$, hence $\max_{a' \in \mathcal{A}} Q_t(b,a') = \max_{a \in \mathcal{A}} Q_t(b, b+a)$, and finally,

$$\max_{a \in \mathcal{A}} Q_t(b,a) = b \cdot c^+ + \max_{a \in \mathcal{A}} Q_t(0,a).$$

In other terms, $V_t(b) = b \cdot c^+ + V_t(0)$. This implies that $V_t(b+1) - V_t(b) \leq c^+$ and V_t is non-decreasing: $(\mathcal{H}_t)$ is true.

Since for all $b \in \mathcal{B}$, $V_N(b) = 0$, $(\mathcal{H}_N)$ is true, and by induction $(\mathcal{H}_t)$ is true for any $t \in \{1, \ldots, N\}$.

The previous induction shows also that for all $t \in \{1, \ldots, N-1\}$:

- $V_t(b) = V_t(0) + b \cdot c^+$,
- for all $b \in \mathcal{B}$, $Q_t(b,a)$ is non-decreasing in a when $a \leq b$,
- for all $b \in \mathcal{B}$, $Q_t(b,a)$ is non-increasing in a when $a \geq b + \alpha_t$.

We now examine the set of maximizers that achieve the maximum value iteration. First, based on lemma 3, the following holds on the relationship between the maximizers in state b and $b+1$

$$\arg\max_{a \in \mathbb{N}} Q_t(b+1, a+1) = \arg\max_{a \in \mathbb{N}} Q_t(b,a),$$

leading by induction to $\arg\max_{a \in \mathbb{N}} Q_t(b, a+b) = \arg\max_{a \in \mathbb{N}} Q_t(0,a)$, *i.e.*

$$\arg\max_{a \in [b, \infty[} Q_t(b,a) = \left\{ a^* + b : a^* \in \arg\max_{a \in \mathbb{N}} Q_t(0,a) \right\}.$$

Then at least one of the maximum of $Q_t(b,a)$ in a is reached in the interval $[b, \infty[$ as $Q_t(b,a)$ is increasing in a in the interval $\{0, \ldots, b\}$, therefore

$$\arg\max_{a \in \mathbb{N}} Q_t(b,a) \supseteq \arg\max_{a \in [b, \infty[} Q_t(b,a),$$

allowing to show that $\pi_t^*(b) \supseteq \{a + b : a \in \pi_t^*(0)\}$, since

$$\arg\max_{a \in \mathbb{N}} Q_t(b,a) \supseteq \left\{ a^* + b : a^* \in \arg\max_{a \in \mathbb{N}} Q_t(0,a) \right\}.$$

□

Remark 2. *Theorem 1 shows in turn that there exists an optimal policy which is monotonic. We can build such a policy by choosing, at each step of the backward induction, the minimum amount of accepted work that maximizes $Q_t(0,a)$ over $a \in \mathcal{A}$ when the batteries are empty, and this amount plus b when the batteries contain b units of energy.*

6.2.3 Bounded Batteries and Workload

We envision now the Ideal-Bounded model, where the batteries keep a maximum capacity, and the amount of work that is available during each period is limited: $w_t \in \mathbb{R}^+$. We stress that the work has to be immediately executed: some work that is proposed but not accepted at a given decision epoch is lost.

The expressions of all $Q_t(b,a)$ remain unchanged but the set of available actions $\mathcal{A}_t$ at decision epoch t is shrinked to $\mathcal{A}_t = \{0, \ldots, w_t\}$, which translates in the value iteration formula: $V_t(b) = \max_{a \in \{0, \ldots, w_t\}} Q_t(b,a)$. This means that lemmas 3 to 5 are still valid in the Ideal-Bounded model.

While following the same pattern of proofs as in the IDEAL-BATTERYBOUNDED model, we need to demonstrate other properties, *e.g.* concavity, on Q_t and V_t in order to exhibit optimal policies. A consequence of adding a constraint on the action set is indeed that the convenient translation formula $V_t(b) = b \cdot c^+ + V_t(0)$ does not hold any more. Instead we show a weaker but true property: V_t is non-decreasingly concave and its rate of change is not greater than c^+.

The basics of the demonstration are shown in the following lemma.

Lemma 6. *If $V_{t+1}(1) - V_{t+1}(0) \leq c^+ + c^-$ and if V_{t+1} is non-decreasingly concave, then $Q_t(0,a)$ is concave in a, and non-increasing when $a \geq \alpha_t$.*

Proof. First of all, lemma 5 shows the variation property ($Q_t(0,a)$ non-increasing in a when $a \geq \alpha_t$). We are interested here in the concavity property.

We define $\Delta_a^2 Q_t(0,a) = \Delta_a Q_t(0,a+1) - \Delta_a Q_t(0,a)$, and for any function ϕ, $\Delta^2\phi(a)$ as $\Delta^2\phi(a) = \Delta\phi(a+1) - \Delta\phi(a)$. Let $t \in \{1,\dots,N-2\}$ and $a \geq 0$. According to the expression of $Q_t(0,a)$ written in lemma 5 ($Q_t(0,a) = c^+ \cdot a - (c^- + c^+) \cdot f(a) + g(a)$, where $f(a) = \sum_{e=0}^{a} q_t^e \cdot (a-e)$ and $g(a) = \sum_{e=0}^{a} q_t^e \cdot V_{t+1}(0) + \sum_{e=a+1}^{B^{\max}+a} q_t^e \cdot V_{t+1}(e-a) + \sum_{e=B^{\max}+a+1}^{\infty} q_t^e \cdot V_{t+1}(B^{\max})$), we have $\Delta^2 f(a) = q_t^{a+1}$. In addition,

$$\begin{aligned}\Delta^2 g(a) = &\sum_{e=a+2}^{B^{\max}+a} q_t^e \cdot (\Delta V_{t+1}(e-a-2) - \Delta V_{t+1}(e-a-1)) \\ &- q_t^{B^{\max}+a+1} \cdot \Delta V_{t+1}(B^{\max}-1) + q_t^{a+1} \cdot \Delta V_{t+1}(0).\end{aligned}$$

This leads to

$$\begin{aligned}\Delta_a^2 Q_t(0,a) &= 0 - (c^+ + c^-) \cdot \Delta^2 f(a) + \Delta^2 g(a) \\ \Delta_a^2 Q_t(0,a) &= q_t^{a+1} \cdot \Big(\Delta V_{t+1}(0) - (c^+ + c^-)\Big) - q_t^{B^{\max}+a+1} \cdot \Delta V_{t+1}(B^{\max}-1) \\ &\quad + \sum_{e=a+2}^{B^{\max}+a} q_t^e \cdot (\Delta V_{t+1}(e-a-2) - \Delta V_{t+1}(e-a-1)).\end{aligned}$$

This implies that if $\Delta V_{t+1}(0) \leq c^+ + c^-$ and V_{t+1} is concave, then $Q_t(0,a)$ is concave in a, hence the lemma. □

For large actions, the behavior of $Q_t(b,a)$ is translated from the shape of $Q_t(0,a)$. We now show that $Q_t(b,a)$ will keep the properties of the previous theorem even for small actions.

Lemma 7. *If $V_{t+1}(1) - V_{t+1}(0) \leq c^+ + c^-$ and if V_{t+1} is non-decreasingly concave, then for all $b \in \mathcal{B}$, $Q_t(b,a)$ is concave in a, non-decreasing when $a \leq b$ and non-increasing when $a \geq \alpha_t$.*

Proof. We assume that $V_{t+1}(1) - V_{t+1}(0) \leq c^+ + c^-$ and V_{t+1} is non-decreasingly concave. Lemmas 3 and 6 show that $Q_t(b,a)$ is concave in a when $a \geq b$, non-decreasing when $a \leq b$ and non-increasing when $a \geq \alpha_t + b$. There remains to prove that

- $Q_t(b,a)$ is concave in a when $a < b$, and
- $\Delta_a Q_t(b,b-1) \geq \Delta_a Q_t(b,b)$ (*i.e.* the junction point between $a < b$ and $a \geq b$ does not break the concavity).

On the one hand, let $a \in \mathcal{A}$ such that $a < b-1$. Equation (6.9) gives that

$$\Delta_a Q_t(b,a) = c^+ - \sum_{e=0}^{a+B^{\max}-b} q_t^e \cdot \Delta V_{t+1}(b-a-1+e),$$

leading by differentiating to

$$\begin{aligned}\Delta_a^2 Q_t(b,a) &= -\sum_{e=0}^{a+1+B^{\max}-b} q_t^e \cdot \Delta V_{t+1}(b-a-2+e) + \sum_{e=0}^{a+B^{\max}-b} q_t^e \cdot \Delta V_{t+1}(b-a-1+e)\\ &= \sum_{e=0}^{a+B^{\max}-b} q_t^e \cdot (\Delta V_{t+1}(b-a-1+e) - \Delta V_{t+1}(b-a-2+e))\\ &\quad - q_t^{a+1+B^{\max}-b} \cdot \Delta V_{t+1}(B^{\max}-1)\\ \Delta_a^2 Q_t(b,a) &= -q_t^{a+1+B^{\max}-b} \cdot \Delta V_{t+1}(B^{\max}-1) + \sum_{e=0}^{a+B^{\max}-b} q_t^e \cdot \Delta^2 V_{t+1}(b-a-2+e)\end{aligned}$$

Since V_{t+1} is non-decreasing and concave by hypothesis, $Q_t(b,a)$ is concave in a when $a < b$.

On the other hand, according to eqs. (6.9) and (6.10), we have

$$\begin{cases} \Delta_a Q_t(b,b-1) = c^+ - \sum_{e=0}^{B^{\max}-1} q_t^e \cdot \Delta V_{t+1}(e) \\ \Delta_a Q_t(b,b) = \Delta_a Q_t(0,0) = c^+ - q_t^0 \cdot (c^+ + c^-) - \sum_{e=1}^{B^{\max}} q_t^e \cdot \Delta V_{t+1}(e-1). \end{cases}$$

Since for all $a \geq b$, $\Delta_a Q_t(b,a) = \Delta_a Q_t(0,a-b)$ (thanks to lemma 3), we have

$$\begin{aligned}\Delta_a Q_t(b,b-1) - \Delta_a Q_t(b,b) &= \Delta_a Q_t(b,b-1) - \Delta_a Q_t(0,0)\\ \Delta_a Q_t(b,b-1) - \Delta_a Q_t(b,b) &= -\sum_{e=1}^{B^{\max}-1} q_t^e \cdot \Delta^2 V_{t+1}(e-1) - q_t^0 \cdot \Delta V_{t+1}(0) + q_t^0 \cdot (c^+ + c^-)\\ &\quad + q_t^{B^{\max}} \cdot \Delta V_{t+1}(B^{\max}-1).\end{aligned}$$

Since $\Delta V_{t+1}(0) \leq c^+ + c^-$, V_{t+1} is concave and V_{t+1} is non-decreasing, we obtain $\Delta_a Q_t(b,b-1) \geq \Delta_a Q_t(b,b)$. □

Theorem 2. *In the* Ideal-Bounded *model, the policy* (π_t) *that fulfills:*

- $\pi_t(0) = \min\left(\arg\max_{a \in \{0,\dots,w_t\}} Q_t(0,a)\right)$,
- *for all* $b \in \{1,\dots,B^{max}\}$, *and* $\pi_t(b) = \min\left(\pi_t(0)+b, w_t\right)$

is optimal.

Proof. In the same way as in the proof of theorem 1, let $(\mathcal{H}_t)$ be the following induction hypothesis: $V_t(b+1) - V_t(b) \leq c^+$ and V_{t+1} is non-decreasingly concave. We assume that for a given $t \in \{1,\dots,N-1\}$, $(\mathcal{H}_{t+1})$ is true and show that $(\mathcal{H}_t)$ is then true.

We call here *action* any a such that $a \in \mathcal{A}$, and qualify this action as valid if $a \in \mathcal{A}_t$, and recall that the set of valid actions $\mathcal{A}_t$ is such that $\mathcal{A}_t = \mathcal{A} \cap [0, w_t]$.

We first look at the situation when the batteries are empty. We denote by $m_{(\text{abs})}$ the smallest value of a that realizes the absolute maximum of $Q_t(0,a)$, that is $m_{(\text{abs})} = \min\left(\arg\max_{a \in \mathbb{N}} Q_t(0,a)\right)$. This action might not be valid, in case there is not enough available work. Still according to lemma 5, we know that $Q_t(0,a)$ is concave in a, therefore $Q_t(0,a)$ is non-decreasing when $a \in \{0,\dots,m_{(\text{abs})}\}$. This implies that the action $a_0^\star = \min(m_{(\text{abs})}, w_t)$, which is valid, is an optimal action (*i.e.* such that $Q_t(0,a_0^\star) = V_t(0)$).

Let $b \in \mathcal{B}$. On the one hand, according to lemma 4, $Q_t(b,a)$ is non-decreasing in a when $a \in \{0,\dots,b\}$. Since lemma 3 shows that $Q_t(b,a) = Q_t(0,a-b)$ for any $a \geq b$, we deduce that

$$\arg\max_{a \in \mathbb{N}} Q_t(b,a) \supseteq \left\{ b + a' : a' \in \arg\max_{a \in \mathbb{N}} Q_t(0,a) \right\},$$

which implies that $b + m_{(\text{abs})} \in \arg\max_{a\in\mathbb{N}} Q_t(b,a)$. On the other hand, we have shown that $Q_t(0,a)$ is non-decreasing in a when $a \in \{0,\dots,m_{(\text{abs})}\}$, hence lemma 3 implies that $Q_t(b,a)$ is non-decreasing in a when $a \in \{b,\dots,b+m_{(\text{abs})}\}$. We then deduce from lemma 4 that $Q_t(b,a)$ is non-decreasing in a when $a \leq b + m_{(\text{abs})}$.

All put together, m_b, defined as $m_b = \min\left(m_{(\text{abs})} + b, w_t\right)$, is valid, and is a maximizer at decision epoch t when there is b amount of energy in the batteries (*i.e.* that maximizes $Q_t(b,a)$ over the set of valid actions).

We need now to show that the property on V_{t+1}, namely non-decreasing concavity, is preserved on V_t. First, lemma 3 implies that

$$\forall (b,a) \in \mathcal{B}\times\mathbb{N},\ Q_t(b,a) = Q_t(B^{\max}, a + B^{\max} - b) + (B^{\max} - b)\cdot c^+.$$

We denote by $b_{(\text{abs})}$ the hypothetical value of batteries for which we hit the workload limit: $b_{(\text{abs})} = w_t - m_{(\text{abs})}$, and we partition $\mathcal{B}$ into at most two parts. Let $\mathcal{B}^+$ and $\mathcal{B}^-$ be defined as: $\mathcal{B}^+ = \mathcal{B}\cap]-\infty, b_{(\text{abs})}]$ and $\mathcal{B}^- = \mathcal{B}\cap [b_{(\text{abs})}, +\infty[$, such that $\mathcal{B} = \mathcal{B}^+ \cup \mathcal{B}^-$, knowing that $\mathcal{B}^+$ or $\mathcal{B}^-$ may be empty. From the first part of the proof we deduce that

$$\begin{cases} m_b = m_{(\text{abs})} + b = w_t - b_{(\text{abs})} + b & \text{if } b \in \mathcal{B}^- \\ m_b = w_t & \text{if } b \in \mathcal{B}^+ \end{cases}, \text{ hence}$$

$$V_t(b) = \begin{cases} Q_t(b, w_t - b_{(\text{abs})} + b) & \text{if } b \in \mathcal{B}^- \\ Q_t(b, w_t) & \text{if } b \in \mathcal{B}^+ \end{cases}$$

$$V_t(b) = \begin{cases} Q_t(B^{\max}, w_t - b_{(\text{abs})} + B^{\max}) - (B^{\max} - b)\cdot c^+ & \text{if } b \in \mathcal{B}^- \\ Q_t(0, w_t - b) + b\cdot c^+ & \text{if } b \in \mathcal{B}^+ \end{cases}$$

We show that V_t is non-decreasingly concave on both $\mathcal{B}^-$ and $\mathcal{B}^+$, and that if V_t spreads over the two sets, *i.e.* $|\mathcal{B}^+| > 1$ and $|\mathcal{B}^-| > 1$ then $\Delta V_t(b_{(\text{abs})} - 1) \leq \Delta V_t(b_{(\text{abs})})$.

First, if $(b, b+1) \in (\mathcal{B}^-)^2$, then

$$\begin{aligned} \Delta V_t(b) &= Q_t(B^{\max}, w_t - b_{(\text{abs})} + B^{\max}) - (B^{\max} - b - 1)\cdot c^+ \\ &\quad - Q_t(B^{\max}, w_t - b_{(\text{abs})} + B^{\max}) + (B^{\max} - b)\cdot c^+ \\ \Delta V_t(b) &= c^+, \end{aligned}$$

which shows that V_t is increasingly concave on $\mathcal{B}^-$, and for all $b \in \mathcal{B}^-$, $\Delta V_t(b) \leq c^+$.

Then, if $(b, b+1) \in (\mathcal{B}^+)^2$,

$$\begin{aligned} \Delta V_t(b) &= Q_t(0, w_t - b - 1) + (b+1)\cdot c^+ - Q_t(0, w_t - b) - b\cdot c^+ \\ \Delta V_t(b) &= -\Delta_a Q_t(0, w_t - b - 1) + c^+. \end{aligned}$$

According to eqs. (6.9) and (6.10), for all $a \in \mathbb{N}$, $\Delta_a Q_t(0,a) \leq c^+$, thus V_t is non-decreasing on $\mathcal{B}^+$. Moreover, by definition of $m_{(\text{abs})}$, $Q_t(0,a)$ is non-decreasing in a when $a \in \{0,\dots,m_{(\text{abs})}\}$. Yet $w_t - b \leq w_t - b_{(\text{abs})} \leq m_{(\text{abs})}$, hence $\Delta_a Q_t(0, w_t - b - 1) \geq 0$, and V_t is non-decreasing on $\mathcal{B}^+$. In addition, if $b + 2 \in \mathcal{B}^+$, $\Delta^2 V_t(b) = \Delta_a^2 Q_t(0, w_t - b - 2)$, thus V_t is concave on $\mathcal{B}^+$.

Finally, if both sides of $b_{(\text{abs})}$ are in the set of valid batteries, *i.e.* $b_{(\text{abs})} + 1 \in \mathcal{B}^+$ and $b_{(\text{abs})} - 1 \in \mathcal{B}^-$,

$$\begin{aligned} \Delta V_t(b_{(\text{abs})} - 1) - \Delta V_t(b_{(\text{abs})}) &= c^+ - (-\Delta_a Q_t(0, w_t - b_{(\text{abs})} - 1) + c^+) \\ \Delta V_t(b_{(\text{abs})} - 1) - \Delta V_t(b_{(\text{abs})}) &= \Delta_a Q_t(0, m_{(\text{abs})} - 1). \end{aligned}$$

This leads to $\Delta V_t(b_{(\text{abs})} - 1) - \Delta V_t(b_{(\text{abs})}) \geq 0$ since $Q_t(0, a)$ is non-decreasing in a when $a \in \{0, \dots, m_{(\text{abs})}\}$.

All put together, V_t is non-decreasingly concave and for all $b \in \{0, \dots, B^{\max} - 1\}$, $\Delta V_t(b) \leq c^+$, and $(\mathcal{H}_t)$ is true.

Since $(\mathcal{H}_N)$ is true, $(\mathcal{H}_t)$ is true for all $t \in \{1, \dots, N\}$, hence the theorem.

□

Remark 3. *One can observe that the optimal actions are always in the domain of the large actions (i.e. $a \geq b$). An optimal policy would then be developed as follows: at each decision epoch, accept as much work as you need to empty the batteries (and as you can, since at most w_t work is available), plus a small amount of work according to the renewable energy production. This is partly due to the ideal efficiency of the batteries: there is no energy loss when the renewable energy transits through the batteries, hence they can be used without moderation. This artefact will disappear in the next section, where the batteries have imperfect efficiency.*

6.3 Non-ideal batteries

6.3.1 Unbounded batteries and workload

Roadmap

As usual, we find an optimal policy and the optimal expectation of the sum of the rewards over all time periods by induction.

In the same line of reasoning as in the previous part, we consider decision epoch t and split the behavior of the MDP into two parts depending on the value of the accepted work a compared to the current state of charge of the batteries b, and we start by expressing $Q_t(b, a)$ in section 6.3.1 as well as all its useful differentiations in section 6.3.1.

We decompose the final theorem 3 into several lemmas, which all assume that the induction hypothesis (that is composed of several properties on V_{t+1}, V'_{t+1} and V''_{t+1}) is fulfilled.

First, we study the impact of the induction hypothesis on Q_t, and show where the extrema along a can be found. In lemmas 8 to 10 we show that for all $b \in \mathcal{B}$, $Q_t(b, a)$ is concave in a and that the extrema along a can be found where the differentiate of Q_t according to a is null.

In lemma 11, for all $b \in \mathcal{B}$ we exhibit an action $m(b)$ among the optimal actions, and show in lemma 12 that variations around a given b cannot change drastically their images by m.

As a consequence, we show that we can differentiate V_t twice, harmlessly, and express it and its differentiates in lemmas 13 and 14.

Finally, Lemmas 15 to 17 prove that the properties on the value iteration and its differentiates (the most important of them being the bounds on the first differentiate of the value iteration and the concavity of V_t) passes from V_{t+1} to V_t through Q_t.

In order not to congest the reasoning with lines of differentiation/integration, we move part of computations details and secondary proofs to chapter A.

Expression of $Q_t(b, a)$

As in the ideal batteries model, $Q_t(b, a)$ follows distinct behaviors depending on the position of a relatively to b, since we cannot lack of energy for the current time period if

$b \geq a$.

Still, we have to ensure that we can integrate and differentiate easily and assume coarse-grain hypotheses on the integrability and differentiability of V_{t+1}: we assume in this section that V_{t+1} is continuous and recall that for all $e \in [E^{\max}, \infty[$, $q_t(e) = 0$.

Case $a \geq b$ The current reward is transformed into the continuous space. For all $(b, a) \in \mathcal{B} \times \mathcal{A}$, $e \mapsto q_t(e) \cdot (a - b - e)$ is integrable on $]0, a - b[$ and

$$r_t(b, a) = a \cdot c^+ - \int_0^{a-b} q_t(e) \cdot (a - b - e) \cdot (c^+ + c^-) de.$$

We integrate by part, integrating $e \mapsto q_t(e)$ and differentiating $e \mapsto (a - b - e)$,

$$r_t(b, a) = a \cdot c^+ - (c^+ + c^-) \cdot \left([F_t(e) \cdot (a - b - e)]_0^{a-b} + \int_0^{a-b} F_t(e) de \right)$$
$$r_t(b, a) = a \cdot c^+ - (c^+ + c^-) \int_0^{a-b} F_t(e) de.$$

We prepare already the continuation by changing variables, with $e \leftarrow e/(a-b)$. We indeed want to integrate between constant bounds (according to a and b), in order to be able to differentiate along a or b without trouble.

$$r_t(b, a) = a \cdot c^+ - (c^+ + c^-) \cdot (a - b) \cdot \int_0^1 F_t((a - b)e) de$$

The contribution of all future steps in Q_t, which is dispatched in V_{t+1}, is partitioned into three parts, such as:

$$Q_t(b, a) = r_t(b, a) + Q_{t,1}(b, a) + Q_{t,2}(b, a) + Q_{t,3}(b, a)$$

.

These partitions allows to isolate different behaviors, which depends on the renewable energy production compared to the taken action a and the state of charge b. We assume first that the whole accepted work can be executed thanks to renewable energy, and denote by $Q_{t,1}(b, a)$ this contribution to the current step. As the batteries may not be ideal, the best option is not to use the energy stored in the batteries, to execute the work with renewable energy sources only, and to store the excess energy in the batteries. From e units of energy produced by renewable energy sources, a units go for the execution of the accepted work, and only $\eta \cdot (e - a)$ units of energy are stored in the batteries due to imperfect efficiency.

V_{t+1} and q_t are continuous on $[0, \infty[$ and for all $e > E^{\max}$, $q_t(e) \cdot V_{t+1}(b + \eta(e - a)) = 0$, hence $e \mapsto q_t(e) \cdot V_{t+1}(b + \eta(e - a))$ is integrable on $]a, \infty[$ and

$$Q_{t,1}(b, a) = \int_a^{\infty} q_t(e) \cdot V_{t+1}(b + \eta(e - a)) de.$$

By changing variables $e \leftarrow e - a$,

$$Q_{t,1}(b, a) = \int_0^{\infty} q_t(a + e) \cdot V_{t+1}(b + \eta e) de.$$

We assume now that the renewable energy production and the energy from the batteries are not enough to execute the accepted work: this happens when the renewable energy

production is less than $a - b$. At the next decision epoch, the batteries are then empty and the contribution to the current step is expressed as

$$Q_{t,2}(b,a) = \int_0^{a-b} q_t(e) \cdot V_{t+1}(0) de = V_{t+1}(0) \cdot F_t(a-b).$$

We assume finally that the total available energy is enough to execute the whole accepted work, but the renewable production is not: the battery needs to be at least partially discharged. Again $e \mapsto q_t(e) \cdot V_{t+1}(b - (a-e))$ is integrable on $]a-b, a[$, and

$$Q_{t,3}(b,a) = \int_{a-b}^{a} q_t(e) \cdot V_{t+1}(b-(a-e)) de$$

We change variables with $e \leftarrow (a-e)/b$ in order to have constant bounds for the integration and obtain

$$Q_{t,3}(b,a) = b \cdot \int_0^1 q_t(a-be) \cdot V_{t+1}(b(1-e)) de$$

All put together,

$$\begin{aligned} Q_t(b,a) = a \cdot c^+ - (c^+ + c^-) \cdot (a-b) \cdot \int_0^1 F_t((a-b)e) de + V_{t+1}(0) \cdot F_t(a-b) \\ + \int_0^\infty q_t(a+e) \cdot V_{t+1}(b+\eta e) de + b \cdot \int_0^1 q_t(a-be) \cdot V_{t+1}(b(1-e)) de \end{aligned} \quad (6.12)$$

Case $a \leq b$ We are sure that there is enough energy stored in the batteries so that the accepted work can be executed during the current time period. The reward is then $r_t(b,a) = a \cdot c^+$. Similarly to the previous case, we partition the contribution of all future rewards into $Q_{t,4}$ and $Q_{t,5}$, depending if there is enough renewable energy production to handle the accepted work or not. In the first case, the contribution is

$$Q_{t,4}(b,a) = \int_a^\infty q_t(e) \cdot V_{t+1}(b + \eta(e-a)) de = \int_0^\infty q_t(a+e) \cdot V_{t+1}(b+\eta e) de,$$

after changing the variable with $e \leftarrow e - a$.

For the lowest energy production, we have

$$Q_{t,5}(b,a) = \int_0^a q_t(e) \cdot V_{t+1}(b-(a-e)) de = a \int_0^1 q_t(ae) \cdot V_{t+1}(b - a(1-e)) de,$$

by changing the variable $e \leftarrow e/a$. Hence

$$Q_t(b,a) = a \cdot c^+ + \int_0^\infty q_t(a+e) \cdot V_{t+1}(b+\eta e) de + a \int_0^1 q_t(ae) \cdot V_{t+1}(b - a(1-e)) de.$$

Differentiations

We are about to differentiate under the integral, hence we need some differentiation properties on q_t and V_{t+1}. By hypothesis, q_t is of class $\mathcal{C}^1$ and for all $e \in [E^{\max}, \infty[$, $q_t(e) = 0$. We assume in addition that V_{t+1} is of class $\mathcal{C}^2$ and V'_{t+1} is bounded (we will see later on that these properties on the value iteration can be passed from $t+1$ to t in the induction proof of theorem 3).

Case $a \geq b$ We compute now the first order differentiates of $Q_t(b,a)$, starting with $\frac{\partial Q_t}{\partial b}(b,a)$. We show that Q_t can be partially differentiated and that this partial differentiates are continuous, which demonstrates that Q_t is differentiable.

We inspect first $r_t(b,a) = a \cdot c^+ - (c^+ + c^-) \cdot (a-b) \cdot \int_0^1 F_t((a-b)e)de$. To simplify the later equations, we define a function f which gives the appropriate value of F_t for a given set of battery state, action and renewable production. Let for all $(b,a,e) \in \mathcal{B} \times \mathcal{A} \times \mathcal{E}$, $f(b,a,e) = F_t((a-b)e)$. On the one hand, for all (b,a), the function $e \mapsto f(b,a,e)$ is differentiable on $\mathcal{E} = [0,\infty[$ and its differentiate along b is $\frac{\partial f}{\partial b}(b,a,e) = -e \cdot q_t((a-b)e)$. We know that for all $e \in [E^{\max}, \infty[$, $q_t(e) = 0$. Let $\phi : [0,\infty[\to \mathbb{R}$ such that

$$\phi(e) = \begin{cases} \max_{e \in [0, E^{\max}]} q_t(e) & \text{if } e < E^{\max} \\ 0 & \text{otherwise} \end{cases} \tag{6.13}$$

For all $(b,a,e) \in \mathcal{B} \times \mathcal{A} \times [0,1]$, $\left|\frac{\partial f}{\partial b}(b,a,e)\right| \leq \phi(e)$. Since ϕ is integrable on $[0,1]$, this implies that we can derive under the integral. r_t is differentiable along b, its differentiate is continuous, and

$$\frac{\partial r_t}{\partial b}(b,a) = (c^+ + c^-) \int_0^1 F_t((a-b)e)de - (c^+ + c^-) \cdot (a-b) \cdot \int_0^1 (-e) q_t((a-b)e)de.$$

We integrate by part the second integral, differentiating $e \mapsto e$ and integrating $e \mapsto (a-b) \cdot q_t((a-b)e)$.

$$\frac{\frac{\partial r_t}{\partial b}(b,a)}{(c^+ + c^-)} = \int_0^1 F_t((a-b)e)de + [eF_t((a-b)e)]_0^1 - \int_0^1 F_t((a-b)e)de = F_t(a-b)$$

We show that we can differentiate under the integral in the expression of $Q_{t,1}(b,a)$. The differentiate domination comes in a similar way: by definition of ϕ in eq. (6.13), we have for all $a \in \mathcal{A}$, $q_t(a+e) \leq \phi(e)$. We know indeed that either $a+e > E^{\max}$ and $q_t(a+e) = 0 \leq \phi(e)$, or $a+e \leq E^{\max}$ (hence $e < E^{\max}$) and $q_t(a+e) \leq \left(\max_{e' \in [0,E^{\max}]} q_t(e')\right) = \phi(e)$. Since by hypothesis there exists K such that for all $b \in \mathcal{B}$, $|V'_{t+1}(b)| \leq K$, we have that for all $(b,a,e) \in \mathcal{B} \times \mathcal{A} \times [0,\infty[$, $\left|q_t(a+e) \cdot V'_{t+1}(b+\eta e)\right| \leq K \cdot \phi(e)$. $e \mapsto K \cdot \phi(e)$ is integrable on $[0,\infty[$ and $(b,a,e) \mapsto q_t(a+e) \cdot V_{t+1}(b+\eta e)$ is of class $\mathcal{C}^1$, hence $Q_{t,1}$ is differentiable along b, its partial differentiate is continuous and

$$\frac{\partial Q_{t,1}}{\partial b}(b,a) = \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b+\eta e)de$$

Straightforwardly,

$$\frac{\partial Q_{t,2}}{\partial b}(b,a) = -V_{t+1}(0) \cdot q_t(a-b)$$

We do not repeat the reasoning, but we claim that under our hypotheses we can differentiate all previously exposed pieces of Q_t under the integral, twice, and obtain continuous first and second order partial differentiates.

$$\begin{aligned} \frac{\partial Q_t}{\partial b}(b,a) = (c^+ + c^-) \cdot F_t(a-b) + \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b+\eta e)de \\ + b \int_0^1 q_t(a-be) \cdot V'_{t+1}(b(1-e))de \end{aligned} \tag{6.14}$$

With the same kind of techniques, we can obtain the first and second order partial differentiates along a

$$\frac{\partial Q_t}{\partial a}(b,a) = c^+ - (c^+ + c^-) \cdot F_t(a-b) \\ - \eta \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b+\eta e)de - b\int_0^1 q_t(a-be) \cdot V'_{t+1}(b(1-e))de \quad (6.15)$$

$$\frac{\partial^2 Q_t}{\partial a^2}(b,a) = \Big(V'_t(0) - (c^+ + c^-)\Big) \cdot q_t(a-b) - (1-\eta) \cdot q_t(a) \cdot V'_{t+1}(b) \quad (6.16)$$

$$+ \eta^2 \int_0^\infty q_t(a+e) \cdot V''_{t+1}(b+\eta e)de + b\int_0^1 q_t(a-be) \cdot V''_{t+1}(b(1-e))de \quad (6.17)$$

Case $a \leq b$ The differentiates of Q_t for small actions are simpler cases, whose formulae are given in the following:

$$\frac{\partial Q_t}{\partial a}(b,a) = c^+ - \eta \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b+\eta e)de - a\int_0^1 q_t(ae) \cdot V'_{t+1}(b-a(1-e))de \quad (6.18)$$

$$\frac{\partial^2 Q_t}{\partial a^2}(b,a) = \eta^2 \int_0^\infty q_t(a+e) \cdot V''_{t+1}(b+\eta e)de + a\int_0^1 q_t(ae) \cdot V''_{t+1}(b-a(1-e))de \quad (6.19)$$

$$\frac{\partial Q_t}{\partial b}(b,a) = \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b+\eta e)de + a\int_0^1 q_t(ae) \cdot V'_{t+1}(b-a(1-e))de \quad (6.20)$$

The other less primordial second order differentiates can be found in section A.1.1.

Properties

We define $(\mathcal{H}_{t+1})$ the induction hypothesis such that

- V_{t+1} is of class $\mathcal{C}^2$ and non-decreasingly concave
- For all $b \in \mathcal{B}$, $V'_{t+1}(b) < \min(c^+/\eta, c^+ + c^-)$

We notice first that if $(\mathcal{H}_{t+1})$ is true, then the hypotheses of the two previous sections (where we compute Q_t and its differentiates) are fulfilled, thus the expressions of Q_t are valid, Q_t is of class $\mathcal{C}^2$ and all its partial differentiates formulae are valid.

Lemma 8. *If $(\mathcal{H}_{t+1})$ is true, then for all $b \in \mathcal{B}$, $a \mapsto Q_t(b,a)$ is concave.*

Proof. If $(\mathcal{H}_{t+1})$ is true, then $V'_{t+1} \geq 0$ and $V''_{t+1} \leq 0$, hence for all $(b,a) \in \mathcal{B} \times \mathcal{A}$, $\frac{\partial^2 Q_t}{\partial a^2}(b,a) \leq 0$ as a sum of non-positive terms. □

Lemma 9. *If $(\mathcal{H}_{t+1})$ is true, then for all $b \in \mathcal{B}$, $\frac{\partial Q_t}{\partial a}(b,0) > 0$.*

Proof. For all $b \in \mathcal{B}$,

$$\begin{aligned}\frac{\partial Q_t}{\partial a}(b,0) &= c^+ - \eta \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b+\eta e)de \\ &> c^+ - \eta \cdot \frac{c^+}{\eta} \int_0^\infty q_t(a+e)de \\ &> c^+ - \eta \cdot \frac{c^+}{\eta} \\ \frac{\partial Q_t}{\partial a}(b,0) &> 0\end{aligned}$$

□

Lemma 10. *If $(\mathcal{H}_{t+1})$ is true, then for all $b \in \mathcal{B}$, there exists $a_n \in \mathcal{A}$ such that $\frac{\partial Q_t}{\partial a}(b,a_n) < 0$.*

Proof. If $(\mathcal{H}_{t+1})$ is true, then for all $(b,a) \in \mathcal{B} \times \mathcal{A}$,

$$\frac{\partial Q_t}{\partial a}(b,a) \leq c^+ - (c^+ + c^-) \cdot F_t(a-b).$$

The lemma comes then from the fact that $F_t(a-b) \xrightarrow[a\to\infty]{} 1$. □

Lemma 11. *If $(\mathcal{H}_{t+1})$ is true, let the function $m : \mathcal{B} \to \mathcal{A}$ defined for any $b \in \mathcal{B}$ as*

$$m(b) = \inf \mathcal{A}_b^\star, \text{ where } \mathcal{A}_b^\star = \left\{ a \in \mathcal{A} : \frac{\partial Q_t}{\partial a}(b,a) = 0 \right\}.$$

For any $b \in \mathcal{B}$, $m(b)$ is a maximizer at decision epoch t if there is b energy in the batteries.

$$V_t(b) = Q_t(b, m(b))$$

Proof. According to lemmas 8 and 9, for any $b \in \mathcal{B}$, $Q_t(b,a) = \max_{a' \in \mathcal{A}}(Q_t(b,a'))$ if and only if $\frac{\partial Q_t}{\partial a}(b,a) = 0$. Since $a \mapsto Q_t(b,a)$ is concave, $\mathcal{A}_b^\star$ is an interval, and as $\frac{\partial Q_t}{\partial a}$ is continuous, this interval is closed. As a result, $\inf \mathcal{A}_b^\star \in \mathcal{A}_b^\star$, hence the lemma. □

Lemma 12. *If $(\mathcal{H}_{t+1})$ is true, the function m is of bounded variation, and*

$$\forall (b,b') \in \mathcal{B}^2, \quad \frac{|m(b) - m(b')|}{|b-b'|} \leq \eta$$

Proof. Let $b \in \mathcal{B}$ and $h \in \mathbb{R}^+$. In order to study the variations of m, we examine where is $m(b+\eta h)$ compared to $m(b)$ and $m(b)+h$. If $m(b+\eta h)$ belongs to $[m(b), m(b)+h]$, we will have proven the lemma. More precisely, we show two propositions: under a state of charge of $b+\eta h$, (i) there exists an optimal action belonging to $[m(b), m(b)+h]$, and (ii) $m(b+\eta h)$ is indeed within the same interval.

On the one hand, we have that

$$\frac{\partial Q_t}{\partial a}(b+\eta h, m(b)+h) = \frac{\partial Q_t}{\partial a}(b+\eta h, m(b)+h) - \frac{\partial Q_t}{\partial a}(b, m(b)),$$

since by definition of $m(b)$, we have $\frac{\partial Q_t}{\partial a}(b, m(b)) = 0$. In section A.1.2, we show that

$$\begin{aligned}
&\frac{\partial Q_t}{\partial a}(b+\eta h, a+h) - \frac{\partial Q_t}{\partial a}(b,a) \\
&= -\int_{a-b}^{a-b+(1-\eta)h} q_t(e)\left((c^+ + c^-) - V'_{t+1}(b+e-a)\right) de \\
&\quad - \int_{a}^{a+h} q_t(e)\cdot\left(V'_{t+1}(b+e-a-(1-\eta)h) - \eta V'_{t+1}(b+\eta(e-a))\right) de \\
&\quad - \int_{a-b+(1-\eta)h}^{a} q_t(e)\cdot\left(V'_{t+1}(b+e-a-(1-\eta)h) - V'_{t+1}(b+e-a)\right) de.
\end{aligned}$$

Primo, for all $b' \in \mathcal{B}$, $V'_{t+1}(b') \leq c^+ + c^-$. *Secundo*, as V'_{t+1} is non-increasing, for all $e \in [a, a+h]$, $b+e-a-(1-\eta)h \leq b+\eta(e-a)$, thus $V'_{t+1}(b+e-a-(1-\eta)h) \geq V'_{t+1}(b+\eta(e-a)) \geq \eta V'_{t+1}(b+\eta(e-a))$. *Tertio*, for the same reason, for all $e \in [a-b+(1-\eta)h, a]$, $b+e-a-(1-\eta)h \leq b+e-a$, hence $V'_{t+1}(b+e-a-(1-\eta)h) - V'_{t+1}(b+e-a)$. All put together, we obtain

$$\frac{\partial Q_t}{\partial a}(b+\eta h, a+h) \leq \frac{\partial Q_t}{\partial a}(b,a), \quad \text{therefore} \quad \frac{\partial Q_t}{\partial a}(b+\eta h, m(b)+h) \leq 0.$$

On the other hand, we have seen that for all $(b,a) \in \mathcal{B}\times\mathcal{A}$, $\frac{\partial^2 Q_t}{\partial b \partial a}(b,a) \geq 0$, which means that

$$\frac{\partial Q_t}{\partial a}(b+\eta h, m(b)) \geq \frac{\partial Q_t}{\partial a}(b, m(b)) = 0$$

Altogether, we have on the left side $\frac{\partial Q_t}{\partial a}(b+\eta h, m(b)+h) \leq 0$, and on the right side $\frac{\partial Q_t}{\partial a}(b+\eta h, m(b)) \geq 0$. We prove now that $m(b+\eta h)$ cannot be less than $m(b)$ or more than $m(b)+h$.

Since $Q_t(b+\eta h, a)$ is concave in a, this implies that for any $h \in \mathbb{R}^+$, there exists $a_h \in [m(b), m(b)+h]$ such that $\frac{\partial Q_t}{\partial a}(b+\eta h, a_h) = 0$. In other words, for any $b' \in [b, \infty[\times\mathcal{B}$, there exists $a' \in [m(b), m(b)+\eta^{-1}(b'-b)]$ such that $\frac{\partial Q_t}{\partial a}(b', a') = 0$. By definition of $m(b')$, which is the smallest action where the first order partial differentiate along a is null, this implies that $m(b') \leq a' \leq m(b) + \eta^{-1}(b'-b)$.

For any $b' \in [b, \infty[\times\mathcal{B}$, could there exist $a \in \mathcal{A}$ such that $a < m(b)$ and $\frac{\partial Q_t}{\partial a}(b', a) = 0$? By definition of $m(b)$ and because $Q_t(b,a)$ is concave in a, for all $a \in \mathcal{A}$, if $a < m(b)$, then $\frac{\partial Q_t}{\partial a}(b,a) > 0$. Since $\frac{\partial Q_t}{\partial a}(b,a)$ is non-decreasing in b, this implies that for all $(b', a) \in \mathcal{B}\times\mathcal{A}$, $\frac{\partial Q_t}{\partial a}(b,a) > 0$ if $a < m(b)$ and $b' > b$. This proves that for all $b' \in \mathcal{B}$, if $b' > b$ then $m(b') \geq m(b)$.

All put together, we have shown that

$$\forall (b, b') \in \mathcal{B}^2, b' > b \Rightarrow m(b) \leq m(b') \leq m(b) + \eta^{-1}(b'-b).$$

In other terms,

$$\forall (b, b') \in \mathcal{B}^2, b' > b \Rightarrow 0 \leq \frac{m(b') - m(b)}{b'-b} \leq \eta^{-1}.$$

We use similar arguments when $b' < b$ to obtain the lemma. □

Remark 4. *We just have shown that*

$$\forall (b, a, h) \in \mathcal{B}\times\mathcal{A}\times\mathbb{R}^+, \quad \frac{\partial Q_t}{\partial a}(b+\eta h, a+h) \leq \frac{\partial Q_t}{\partial a}(b,a). \tag{6.21}$$

This property gives insight about the geometry of the problem when there are several optimal action for a given b. The non-uniqueness of optimal actions translates into the existence of $b \in \mathcal{B}$ and of a non-singleton interval $I \subset \mathcal{A}$ such that for all $a \in I$, $\frac{\partial Q_t}{\partial a}(b,a) = 0$. We show in section A.1.3 that there exists then a subset of $\mathcal{B} \times \mathcal{A}$ containing $\{b\} \times I$ with a non-null area where any action is optimal.

Lemma 13. *If $(\mathcal{H}_{t+1})$ is true then V_t is differentiable and*

$$V_t'(b) = \frac{\partial Q_t}{\partial b}(b, m(b)). \tag{6.22}$$

Proof. Let $\varepsilon \in \mathbb{R}_\star^+$ and let $b \in \mathcal{B}$. We want to show that there exists $r \in \mathbb{R}_\star^+$ such that

$$\forall b' \in \mathcal{B}, |b - b'| < r \Rightarrow \left| \frac{Q_t(b, m(b)) - Q_t(b', m(b'))}{b - b'} - \frac{\partial Q_t}{\partial b}(b, m(b)) \right| < \varepsilon.$$

We can first apply the decomposition that will drive the proof. We observe that for all $b' \in \mathcal{B}^2$,

$$\begin{aligned}
&\left| \frac{Q_t(b, m(b)) - Q_t(b', m(b'))}{b - b'} - \frac{\partial Q_t}{\partial b}(b, m(b)) \right| \\
&< \left| \frac{Q_t(b, m(b)) - Q_t(b', m(b))}{b - b'} + \frac{Q_t(b', m(b)) - Q_t(b', m(b'))}{b - b'} - \frac{\partial Q_t}{\partial b}(b, m(b)) \right| \\
&< \left| \frac{Q_t(b, m(b)) - Q_t(b', m(b))}{b - b'} - \frac{\partial Q_t}{\partial b}(b, m(b)) \right| + \left| \frac{Q_t(b', m(b)) - Q_t(b', m(b'))}{b - b'} \right| \\
&< \left| \frac{Q_t(b, m(b)) - Q_t(b', m(b))}{b - b'} - \frac{\partial Q_t}{\partial b}(b, m(b)) \right| \\
&\quad + \left| \frac{Q_t(b', m(b)) - Q_t(b', m(b'))}{m(b) - m(b')} \right| \times \left| \frac{m(b) - m(b')}{b - b'} \right|.
\end{aligned}$$

Q_t is differentiable, hence for all $(b', a) \in \mathcal{B} \times \mathcal{A}$ there exists r_1 such that

$$\forall a' \in \mathcal{A}, |a - a'| < r_1 \Rightarrow \left| \frac{Q_t(b', a) - Q_t(b', a')}{a - a'} - \frac{\partial Q_t}{\partial a}(b', a) \right| < \frac{\eta\varepsilon}{2}.$$

Especially, for all $b' \in \mathcal{B}$,

$$\begin{aligned}
\forall a' \in \mathcal{A}, |m(b') - a'| < r_1 \Rightarrow \left| \frac{Q_t(b', m(b')) - Q_t(b', a')}{m(b') - a'} - \frac{\partial Q_t}{\partial a}(b', m(b')) \right| < \frac{\eta\varepsilon}{2} \\
\Rightarrow \left| \frac{Q_t(b', m(b')) - Q_t(b', a')}{m(b') - a'} \right| < \frac{\eta\varepsilon}{2},
\end{aligned}$$

as $\frac{\partial Q_t}{\partial a}(b', m(b')) = 0$, by definition of the maximizer m. According to lemma 12,

$$\forall (b_1, b_2) \in \mathcal{B}^2, \frac{|m(b_1) - m(b_2)|}{|b_1 - b_2|} \leq \eta^{-1},$$

which leads to

$$\begin{aligned}
\forall (b_1, b_2) \in \mathcal{B}^2, |b_1 - b_2| < \eta \cdot r_1 \quad &\Rightarrow \frac{|m(b_1) - m(b_2)|}{\eta \cdot r_1} < \frac{|m(b_1) - m(b_2)|}{|b_1 - b_2|} \leq \eta^{-1} \\
&\Rightarrow |m(b_1) - m(b_2)| < r_1.
\end{aligned}$$

Altogether,

$$\begin{aligned}\forall b' \in \mathcal{B}, |b-b'| < \eta \cdot r_1 \Rightarrow \big|m(b) - m(b')\big| &< r_1 \\ \Rightarrow \left|\frac{Q_t(b', m(b')) - Q_t(b', m(b))}{m(b') - m(b)}\right| &< \frac{\eta\varepsilon}{2}.\end{aligned}$$

Lastly, Q_t is differentiable, hence for all $a \in \mathcal{B}$ there exists r_2 such that

$$\forall b' \in \mathcal{B}, |b-b'| < r_2 \Rightarrow \left|\frac{Q_t(b', a) - Q_t(b, a)}{b-b'} - \frac{\partial Q_t}{\partial b}(b, a)\right| < \frac{\varepsilon}{2},$$

and in particular,

$$\forall b' \in \mathcal{B}, |b-b'| < r_2 \Rightarrow \left|\frac{Q_t(b', m(b)) - Q_t(b, m(b))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))\right| < \frac{\varepsilon}{2}.$$

Finally, for all $b' \in \mathcal{B}$,

$$\begin{aligned}|b-b'| < \min(\eta r_1, r_2) \Rightarrow \left|\frac{Q_t(b, m(b)) - Q_t(b', m(b'))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))\right| &< \frac{\varepsilon}{2} + \frac{\eta\varepsilon}{2} \times \eta^{-1} \\ \Rightarrow \left|\frac{Q_t(b, m(b)) - Q_t(b', m(b'))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))\right| &< \varepsilon.\end{aligned}$$

This holds for any $\varepsilon \in \mathbb{R}_\star^+$, and for any $b \in \mathcal{B}$, hence V_t is differentiable in any point of $\mathcal{B}$, and

$$\forall b \in \mathcal{B}, \quad V_t'(b) = \frac{\partial Q_t}{\partial b}(b, m(b))$$

□

Lemma 14. *If $(\mathcal{H}_{t+1})$ is true then V_t is of class $\mathcal{C}^2$ and, for all $b \in \mathcal{B}$,*

- *if $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) \neq 0$ then m is differentiable in b and*

$$V_t''(b) = \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) + m'(b) \cdot \frac{\partial^2 Q_t}{\partial b \partial a}(b, m(b)) \tag{6.23}$$

- *if $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) = 0$ then*

$$V_t''(b) = \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) \tag{6.24}$$

Proof. We first prove that V_t' is differentiable. Let $b \in \mathcal{B}$. We show that either $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) \neq 0$ and we can apply the implicit function theorem to prove that $m \in \mathcal{C}^1$ and differentiate V_t' without trouble; or $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) = 0$ and we can show that V_t' is differentiable and compute V_t'' in the same way as in lemma 13.

We assume first that $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) \neq 0$. Since $\frac{\partial^2 Q_t}{\partial a^2}$ is continuous, there exists an open neighborhood U of $(b, m(b))$ such that for all $(b, a) \in U$, $\frac{\partial^2 Q_t}{\partial a^2}(b, a) \neq 0$. In addition, by definition, $\frac{\partial Q_t}{\partial a}(b, m(b)) = 0$, and $\frac{\partial Q_t}{\partial a} \in \mathcal{C}^1$. According to the implicit function theorem, there exists a neighborhood W of b, a neighborhood $U' \subseteq U$ of $(b, m(b))$ and a function $\psi : \mathcal{B} \to \mathcal{A}$ such that $\psi \in \mathcal{C}^1$ and for all $(b', a') \in \mathcal{B} \times \mathcal{A}$

$$(b', a') \in U' \ \wedge \ \frac{\partial Q_t}{\partial a}(b', a') = 0 \quad \Leftrightarrow \quad b' \in W \ \wedge \ a' = \psi(b').$$

But for all $(b', a') \in U'$, we know that $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) \neq 0$, which implies that $m(b')$ is the unique solution of $\frac{\partial Q_t}{\partial a}(b', a) = 0$. In other words, for all $(b', a') \in \mathcal{B} \times \mathcal{A}$, $\psi(b') = m(b')$. Therefore m is in $\mathcal{C}^1$ within an open neighborhood of b.

We can then apply the classical differentiation formula and obtain

$$V_t''(b) = \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) + m'(b) \cdot \frac{\partial^2 Q_t}{\partial a \partial b}(b, m(b)).$$

The implicit function theorem also stipulates that

$$m'(b) = -\frac{\frac{\partial^2 Q_t}{\partial a \partial b}(b, m(b))}{\frac{\partial^2 Q_t}{\partial a^2}(b, m(b))}.$$

We assume now that $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) = 0$. We notice first in eqs. (6.16) and (6.19) that, since all sumed terms in $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b))$ are non-positive terms, all these terms are null, hence $\frac{\partial^2 Q_t}{\partial b \partial a}(b, m(b)) = 0$. We first split the initial expression to exhibit the direction of the proof:

$$\begin{aligned}
&\left| \frac{\frac{Q_t(b,m(b)) - Q_t(b',m(b'))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))}{b - b'} - \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) \right| \\
< &\left| \frac{\frac{Q_t(b,m(b)) - Q_t(b',m(b)) + Q_t(b',m(b)) - Q_t(b',m(b'))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))}{b - b'} - \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) \right| \\
< &\left| \frac{\frac{Q_t(b,m(b)) - Q_t(b',m(b))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))}{b - b'} - \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) \right| + \left| \frac{Q_t(b', m(b)) - Q_t(b', m(b'))}{(b-b')^2} \right| \\
< &\left| \frac{\frac{Q_t(b,m(b)) - Q_t(b',m(b))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))}{b - b'} - \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) \right| \\
&+ \left| \frac{Q_t(b', m(b)) - Q_t(b', m(b'))}{(m(b) - m(b'))^2} \left(\frac{m(b) - m(b')}{b - b'} \right)^2 \right|.
\end{aligned}$$

We show in section A.1.4 that there exists $r \in \mathbb{R}_\star^+$ such that for all $b' \in \mathcal{B}$,

$$|b - b'| < r \Rightarrow \left| \frac{\frac{Q_t(b,m(b)) - Q_t(b',m(b'))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))}{b - b'} - \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) \right| < \varepsilon.$$

We prove in a similar way that V_t'' is continuous. Let $b \in \mathcal{B}$. If $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) \neq 0$, there exists a neighborhood around $(b, m(b))$ where $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) \neq 0$, *i.e.* where $V_t''(b) = \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) + m'(b) \cdot \frac{\partial^2 Q_t}{\partial a \partial b}(b, m(b))$. All functions are continuous around b or around $(b, m(b))$, hence V_t'' is continuous in b.

If $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) = 0$, $V_t''(b) = \frac{\partial^2 Q_t}{\partial b^2}(b, m(b))$. We can find a neighborhood of b such that for all point b' in this neighborhood, $\frac{\partial^2 Q_t}{\partial a^2}(b', m(b'))$ is as close as we want to $\frac{\partial^2 Q_t}{\partial b^2}(b, m(b))$, hence all $V_t''(b')$ such that $\frac{\partial^2 Q_t}{\partial a^2}(b', m(b')) = 0$ are as close as we want to $V_t''(b)$. Moreover, we can also find a neighborhood of b such that for all point b' in this neighborhood, $\frac{\partial^2 Q_t}{\partial a \partial b}(b', m(b'))$ is as close as we want to $\frac{\partial^2 Q_t}{\partial a^2}(b)$b,$m(b)$ which is null since $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) = 0$. Since m is of bounded variations, it implies that we can find a neighborhood of b such that for all point b' in this neighborhood, all $V_t''(b')$ such that $\frac{\partial^2 Q_t}{\partial a^2}(b', m(b')) \neq 0$ are as close as we want to $V_t''(b)$. Altogether, V_t'' is continuous in b. □

Lemma 15. *If $(\mathcal{H}_{t+1})$ is true, then V_t is concave.*

Proof. We show that for all $b_0 \in \mathcal{B}$, $V_t''(b_0) \leq 0$. If $\frac{\partial^2 Q_t}{\partial a^2}(b_0, m(b_0)) = 0$, then $V_t''(b) = \frac{\partial^2 Q_t}{\partial b^2}(b_0, m(b_0)) \leq 0$, since $V_{t+1}'(0) < c^+ + c^-$ and V_{t+1}'' is concave.

If $\frac{\partial^2 Q_t}{\partial a^2}(b_0, m(b_0)) \neq 0$, then $V_t''(b_0) = \frac{\partial^2 Q_t}{\partial b^2}(b_0, m(b_0)) + m'(b_0)\frac{\partial^2 Q_t}{\partial b \partial a}(b_0, m(b_0))$. This expression alone is not enough to conclude: we show in section A.1.5 how to overcome this issue by deriving

$$\begin{aligned} V_t''(b_0) = &-(1-\eta)\cdot q_t(a)\cdot V_{t+1}'(b_0)\cdot m'(b_0) \\ &+ (1-\eta)(1-\eta\cdot m'(b_0))\cdot \int_0^\infty q_t(a+e)\cdot V_{t+1}''(b_0+\eta e)de \end{aligned}$$

Since V_{t+1} is non-decreasingly concave (by the induction hypothesis $(\mathcal{H}_{t+1})$), $\eta \leq 1$ (by hypothesis), and $m'(b_0) \in [0, \eta^{-1}]$ (thanks to lemma 12), we have $V_t''(b_0) \leq 0$, hence the lemma. □

Lemma 16. *If $(\mathcal{H}_{t+1})$ is true then for all $b \in \mathcal{B}$, $V_t'(b) < c^+\eta^{-1}$.*

Proof. We prove that $V_t'(0) < c^+\eta^{-1}$, which shows by concavity of V_t (proved in lemma 15) that for all $b \in \mathcal{B}$, $V_t'(b) < c^+\eta^{-1}$.

By definition of $m(0)$, $\frac{\partial Q_t}{\partial a}(0, m(0)) = 0$. As a consequence,

$$(c^+ + c^-)\cdot F_t(m(0)) = c^+ - \eta\int_0^\infty q_t(m(0)+e)\cdot V_{t+1}'(\eta e)de, \quad \text{hence}$$

$$\begin{aligned} V_t'(0) &= \frac{\partial Q_t}{\partial b}(0, m(0)) \\ &= (c^+ + c^-)\cdot F_t(m(0)) + \int_0^\infty q_t(m(0)+e)\cdot V_{t+1}'(b+\eta e)de \\ V_t'(0) &= c^+ + (1-\eta)\int_0^\infty q_t(m(0)+e)\cdot V_{t+1}'(b+\eta e)de. \end{aligned}$$

By hypothesis, for all $b \in \mathcal{B}$, $V_{t+1}'(b) < c^+\eta^{-1}$, thus

$$\begin{aligned} V_t'(0) &< c^+ + (1-\eta)\frac{c^+}{\eta}\int_0^\infty q_t(m(0)+e)de \\ &< c^+ + \frac{(1-\eta)\cdot c^+}{\eta} \\ V_t'(0) &< \frac{c^+}{\eta}. \end{aligned}$$

□

Lemma 17. *If $(\mathcal{H}_{t+1})$ is true then for all $b \in \mathcal{B}$, $V_t'(b) < c^+ + c^-$.*

Proof. We show that $V_t'(0) < c^+ + c^-$. Since by hypothesis, for all $b \in \mathcal{B}$, $V_{t+1}'(b) < c^+ + c^-$, and thanks to lemma 13,

$$\begin{aligned} V_t'(0) &= \frac{\partial Q_t}{\partial b}(0, m(0)) \\ &= (c^+ + c^-)\cdot F_t(m(0)) + \int_{m(0)}^\infty q_t(e)\cdot V_{t+1}'(b+\eta(e-m(0)))de \\ &< (c^+ + c^-)\cdot \int_0^{m(0)} q_t(e)de + (c^+ + c^-)\cdot \int_{m(0)}^\infty q_t(e)de \\ V_t'(0) &< c^+ + c^-. \end{aligned}$$

□

We gather all lemmas into the following main theorem.

Theorem 3. *In the* Imperfect-Unbounded *model, the policy* $(\pi_t)_{t\in\{1,\dots,N-1\}}$ *that fulfills* $\pi_t(b) = \inf \mathcal{A}_b^\star$*, where*

$$\begin{aligned}\mathcal{A}_b^\star = \Big\{a \geq b : c^+ - (c^+ + c^-)\cdot F_t(a-b) - \eta\int_0^\infty q_t(a+e)\cdot V'_{t+1}(b+\eta e)de \\ - b\int_0^1 q_t(a-be)\cdot V'_{t+1}(b(1-e))de = 0\Big\} \\ \cup \Big\{a \leq b : \int_0^\infty q_t(a+e)\cdot V'_{t+1}(b+\eta e)de + a\int_0^1 q_t(ae)\cdot V'_{t+1}(b-a(1-e))de = 0\Big\}\end{aligned}$$

is optimal. In addition, for all $b \in \mathcal{B}$*, we can express* V'_t *easily*

$$V'_t(b) = c^+ + (1-\eta)\cdot\int_0^\infty q_t(\pi_t(b)+e)\cdot V'_{t+1}(b+\eta e)de.$$

Proof. Following all lemmas, the induction proof on the optimal policy is trivial when noticing that for all $b \in \mathcal{B}$, $V_N(b) = 0$, implying that V_N is of class $\mathcal{C}^2$ and $0 \leq V_N < \min(c^+ + c^-, c^+/\eta)$.

The expression of V'_t comes from the translation of the fact that $\frac{\partial Q_t}{\partial a}(b, \pi_t(b)) = 0$ into V'_t defined as $V'_t = \frac{\partial Q_t}{\partial b}(b, \pi_t(b))$ in lemma 13.

If $\pi_t(b) \geq b$, then according to eq. (6.15),

$$\begin{aligned}(c^+ + c^-)\cdot F_t(\pi_t(b)-b) + b\int_0^1 q_t(\pi_t(b)-be)\cdot V'_{t+1}(b(1-e))de \\ = c^+ - \eta\int_0^\infty q_t(\pi_t(b)+e)\cdot V'_{t+1}(b+\eta e)de,\end{aligned}$$

and according to eqs. (6.14) and (6.22),

$$\begin{aligned}V'_t(b) = (c^+ + c^-)\cdot F_t(\pi_t(b)-b) + \int_0^\infty q_t(\pi_t(b)+e)\cdot V'_{t+1}(b+\eta e)de \\ + b\int_0^1 q_t(\pi_t(b)-be)\cdot V'_{t+1}(b(1-e))de,\end{aligned}$$

from which we obtain

$$V'_t(b) = c^+ + (1-\eta)\cdot\int_0^\infty q_t(\pi_t(b)+e)\cdot V'_{t+1}(b+\eta e)de.$$

We show in section A.1.6 that the case $\pi_t(b) \leq b$ results in the same expression for V'_t, hence the theorem. □

Bounded battery, unbounded workload

The structure, *i.e.* the lemma decomposition, that has lead to the expression of an optimal policy, together with the total expected reward, in the Imperfect-Unbounded model can be reused in the Imperfect-BatteryBounded model, where the battery has a maximum capacity.

In section A.2 we exhibit the impact of the maximum battery capacity on the expressions of Q_t and its differentiates. From these formulae we can deduce that all lemmas hold:

- $Q_t(b, \cdot)$ is concave,

- its maxima along a for a given b can be found where its partial differentiate according to a is null,
- the policy that chooses the smallest accepted work that achieves the maximum is of bounded variations,
- V_t is of class $\mathcal{C}^2$, V_t' is still bounded and the same bounds hold.

The following theorem can then be demonstrated with the same lemmas.

Theorem 4. *In the* Imperfect-BatteryBounded *model, the policy* $(\pi_t)_{t\in\{1,\dots,N-1\}}$ *that fulfills* $\pi_t(b) = \inf \mathcal{A}_b^\star$*, where*

$$\begin{aligned}\mathcal{A}_b^\star = \Big\{ a \geq b : c^+ - (c^+ + c^-)\cdot F_t(a-b) - \int_{a-b}^{a} q_t(e)\cdot V'_{t+1}(b+e-a)de \\ - \eta \int_{a}^{a+\eta^{-1}(B^{max}-b)} q_t(e) V'_{t+1}(b+(e-a)\eta)de = 0 \Big\} \\ \cup \Big\{ a \leq b : c^+ + \int_{a}^{a+\eta^{-1}(B^{max}-b)} q_t(e) V'_{t+1}(b+(e-a)\eta)de \\ - \int_0^a q_t(e)\cdot V'_{t+1}(b+e-a)de = 0 \Big\}\end{aligned}$$

is optimal. In addition, for all $b \in \mathcal{B}$*, we have a compact expression for* V_t'

$$V_t'(b) = c^+ + \frac{1-\eta}{\eta}(B^{max}-b)\cdot \int_0^1 q_t(\eta^{-1}(B^{max}-b)e + \pi_t(b)) V'_{t+1}(b-(B^{max}-b)e)de.$$

Bounded battery and workload

We consider now the Imperfect-Bounded model, where the workload that is submitted to the datacenter is finite, *i.e.* for all $t \in \{1,\dots,N-1\}$, $w_t \in \mathbb{R}^+$. This model is at the crossroads between Ideal-Bounded and Imperfect-BatteryBounded. Compared to Ideal-Bounded, the energy losses during the charges and discharges of the battery do not affect the structure of the problem, and we can use the same technique that we employed in the Ideal-bounded model, where the optimal action for a given state of charge was the minimum between the available workload w_t and the optimal action as if there was no limit on the accepted work.

We give here the sketch of the reasoning. As we did since the beginning, we perform the proof by induction. First, we define what shall appear in the induction hypothesis $(\mathcal{H}_{t+1})$. Because of the workload limit, $(\mathcal{H}_{t+1})$ slightly differs from the induction hypothesis used in the Imperfect-BatteryBounded model. Especially, the regularity of V_t loosens when the optimal action with no consideration of the workload hits the workload wall (in case we would like to execute more work than available). The induction hypothesis includes that V_t is of class $\mathcal{C}^1$ and piecewise $\mathcal{C}^2$, instead of being of class $\mathcal{C}^2$. The other hypotheses (V_{t+1} non-decreasingly concave and the bounds on V'_{t+1}) remain in $(\mathcal{H}_{t+1})$.

Next, despite V_{t+1} not being of class $\mathcal{C}^2$, Q_t is still in $\mathcal{C}^2$. This can be shown by splitting the integrals that appears in Q_t to avoid the points where V_t' may not be differentiable. This does not add additional terms, hence Q_t can be considered as unchanged compared to the unbounded workload problem.

Then, we can find the optimal actions on the unbounded workload problem by exploring where the differentiate of Q_t according to a nullifies. Within the interval of batteries where the chosen optimal action is not greater than w_t, it constitutes an optimal action of the bounded workload problem. Since for all $b \in \mathcal{B}$, $Q_t(b,a)$ is concave in a, if this action is greater than w_t, then an optimal action of the unbounded workload problem is w_t. For

greater battery states of charge, this action is still optimal, which implies a constant policy in this second interval. V_t is then of class $\mathcal{C}^2$ on both intervals, and on both intervals, if we name π_t the policy described above

$$\forall b \in \mathcal{B}, \quad V_t'(b) = \frac{\partial Q_t}{\partial b}(b, \pi_t(b)).$$

Since both π_t and $\frac{\partial Q_t}{\partial b}$ are continuous, V_t is of class $\mathcal{C}^1$.

Finally, since Q_t is unchanged, the growth and concavity of V_{t+1} pass to V_t and the bounds on V_{t+1}' reverberate on V_t'. This leads to the following theorem.

Theorem 5. *In the* Imperfect-Bounded *model, the policy* $(\pi_t)_{t\in\{1,\dots,N-1\}}$ *that fulfills* $\pi_t(b) = \min(w_t, \inf \mathcal{A}_b^\star)$*, where*

$$\begin{aligned}\mathcal{A}_b^\star = \Big\{ a \geq b : c^+ - (c^+ + c^-) \cdot F_t(a-b) - \int_{a-b}^{a} q_t(e) \cdot V_{t+1}'(b+e-a)de \\ - \eta \int_a^{a+\eta^{-1}(B^{max}-b)} q_t(e) V_{t+1}'(b+(e-a)\eta)de = 0 \Big\} \\ \cup \Big\{ a \leq b : c^+ - \int_0^a q_t(e) \cdot V_{t+1}'(b+e-a)de \\ + \int_a^{a+\eta^{-1}(B^{max}-b)} q_t(e) V_{t+1}'(b+(e-a)\eta)de = 0 \Big\}\end{aligned}$$

is optimal.

6.4 Numerical application and experiments

In the previous section, a complete study of optimal policies in the Imperfect-Bounded problem has been detailed. States and actions being defined in continuous domains, no generic method exists to compute the optimal policy. Based on its identified structure, it is nonetheless possible to design an algorithm to find an approximation of the expected reward (and an associated policy) as close as we desire to the optimal expected reward.

As always, the proof goes by induction: we assume that there exists an approximation V_{t+1}^{app} of the value iteration at decision epoch $t+1$ within a given error margin, and build an approximation of the value iteration V_t^{algo} of the value iteration at decision epoch t within a new error margin. There are four natures of errors when it comes to the approximation of the problem:

- the initial error from the next decision epoch,
- the operated discretizations in order to compute the integrals that appear in Q_t (integral on $e \mapsto q_t(e) \cdot e$ in the reward and integrals on $q_t \cdot V_{t+1}$),
- the margin when computing the maximum on a of Q_t for a given b,
- the last linearization of the value iteration.

Hence, the final maximum error of the value iteration approximation V_t^{app} computed for epoch t is the sum of the margin caused by all these sources. We start with theoretical considerations, assuming that we have discretizations of any function, and separate clearly the independent errors that can sum up. After detailing optimal policies without RES uncertainty information, the methodology for an experimental study is detailed. This methodology also discuss some more practical aspects of implementing the approximation with bounded errors. Simulation results under various conditions are finally presented, followed by an interpretation and discussion of the findings.

6.4.1 Error propagation

We assume that there exists a linear approximation V_{t+1}^{app} and ε_{t+1} such that $|V_{t+1}^{app} - V_{t+1}| < \varepsilon_{t+1}$, and that the cumulative distribution function F_t of renewable energy produced along epoch t is known perfectly, or at least within a negligible bound.

Impact of the initial error We can notice that denoting by Q_t^{app} the function that is obtained after replacing V_{t+1} by V_{t+1}^{app} in $Q_t(b,a)$, we have $|Q_t^{app} - Q_t| < \varepsilon_{t+1}$. We try now to compute the integrals that appears in Q_t^{app} in a discretized way.

Computing the integrals We note that Q_t^{app} is composed of two main parts: the reward for the current decision epoch, and the future rewards.

We start with the reward, once simplified and integrated by part as in section 6.3.1:

$$r_t(b,a) = \begin{cases} a \cdot c^+ - (c^+ + c^-) \int_0^{a-b} F_t(e)de & \text{if a > b} \\ a \cdot c^+ & \text{otherwise} \end{cases}$$

Its value for the case $a \leq b$ is obtained exactly, and we focus on the case $a > b$. As F_t is the cumulative distribution function of the renewable production, it has several properties that helps integrating it numerically: F_t is non-decreasing, with value in $[0,1]$ and $F_t(E^{\max}) = 1$. A simple midpoint rule quadrature with appropriate strategy for selecting the interpolation values can give efficiently an approximation with an error bounded by arbitrarily small $\varepsilon^{(int)}$. The maximum error of a part (rectangle approximating the integral between $[e_0, e_1]$) is no more than:

$$\frac{(F_t(e_1) - F_t(e_0))(e_1 - e_0)}{2}$$

As F_t is non-decreasing and $e_0 < e_1$, this is equals to 0 if and only if $F_t(e_1) = F_t(e_0)$. If the maximal error is non-nul, chosing an intermediate point $e_m = (e_1 + e_0)/2$ and summing the two resulting rectangle (between $[e_0, e_m]$ and $[e_m, e_1]$) is guaranteed to reduce the total maximum error. Hence, by dividing progressively the range $[0, a-b]$ in order to reduce the maximum approximation error each time, we effectively approximate $\int_0^{a-b} F_t(e)de$ for any desired accuracy $\varepsilon^{(int)}$.

We denote $r_t^{alg}(t)$ the $\varepsilon^{(1)}$-approximation of r_t, obtained by computing the integral with a maximal error $\varepsilon^{(int)} = \frac{\varepsilon^{(1)}}{c^+ + c^-}$ when $a > b$.

The future reward that appears in Q_t^{app}, denoted by $P_t(b,a)$, can be $\varepsilon^{(2)}$-approximated thanks to a piecewise constant discretization of V_{t+1}^{app}, as shown in section A.3: we denote this approximate function by ℓ_t^{alg}.

After applying a similar approximation when $a \leq b$, and defining $Q_t^{algo}(b,a) = r_t^{alg}(b,a) + \ell_t^{alg}(b,a)$, we obtain

$$|Q_t^{app} - Q_t^{algo}| \leq |r_t^{alg}(b,a) - r_t(b,a)| + |P_t(b,a) - \ell_t^{alg}(b,a)| < \varepsilon^{(1)} + \varepsilon^{(2)}.$$

To the approximated value iteration The function Q_t^{algo} can then be used to find an approximated value iteration V_t^{algo} by finding for each b the maximum of $Q_t^{algo}(b,a)$ among a. Finding numerically the maximum of a continuous function adds a new error denoted $\varepsilon^{(3)}$: for all b, $|V_t^{algo}(b) - \max_a Q_t^{algo}(b,a)| < \varepsilon^{(3)}$. During the numerical computation of V_t^{algo}, we do not know how the potential error is distributed. Hence, the concavity of the approximated function Q_t^{app} is not guaranteed in Q_t^{algo}. To use efficient maximum finding

algorithm, such as Newtonian methods, it is possible to adapt the sampling method. Considering a set of sample points at a bounded distance $\varepsilon^{(1)} + \varepsilon^{(2)}$ of a concave function (Q_t^{app}), it is always possible to build a concave approximation by keeping a subset of the sample points. Using this property, a custom maximum point algorithm can be elaborated, which converges quickly near the maximum of $Q_t^{app} \pm \varepsilon^{(1)} + \varepsilon^{(2)}$. However, such an iterative method adds necessarily an additional error, denoted $\varepsilon^{(3)}$, which depends on the maximum number of iterations or on the converging criterion tolerance.

Finally, we can build a linear approximation V_t^{lin} of V_t^{algo}, submitted to the last error: $|V_t^{lin} - V_t^{algo}| < \varepsilon^{(4)}$. This maximum error can be set arbitrary low by computing enough points of V_t^{algo}, such as the linear interpolation $V_t^{lin}b$ between two known values at points $b_0 < b_1$ is as close as desired from $V_t^{algo}(b) \forall b \in [b_0, b_1]$.

Total error bound There remains to gather all the error bounds to obtain the final error that will be given to the previous decision epoch.

$$\begin{aligned} |V_t^{lin}(b) - V_t(b)| &\leq |V_t^{lin}(b) - V_t^{algo}(b)| + |V_t^{algo}(b) - V_t(b)| \\ &< \varepsilon^{(4)} + |V_t^{algo}(b) - \max_a Q_t^{algo}(b,a)| + |\max_a Q_t^{algo}(b,a) - \max_a Q_t(b,a)| \end{aligned}$$

We know that

$$|Q_t^{algo}(b,a) - Q_t(b,a)| \leq |Q_t^{algo}(b,a) - Q_t^{app}b, a| + |Q_t^{app}b, a - Q_t(b,a)| < \varepsilon^{(1)} + \varepsilon^{(2)} + \varepsilon_{t+1},$$

hence $|\max_a Q_t^{algo}(b,a) - \max_a Q_t(b,a)| < \varepsilon^{(1)} + \varepsilon^{(2)} + \varepsilon_{t+1}$, and finally,

$$|V_t^{lin}(b) - V_t(b)| < \varepsilon_{t+1} + \sum_{i=1}^{4} \varepsilon^{(i)}.$$

In other words, we have computed a linear approximation V_t^{lin} of V_t with an error margin of $\varepsilon_t = \varepsilon_{t+1} + \sum_{i=1}^{4} \varepsilon^{(i)}$.

6.4.2 Optimal policy without uncertainty information

To study how taking uncertainties into account can improve the decisions, an optimal policy for deterministic renewable production forecast is given. Hence the only difference with the model considered by our approach is the energy available at a time t, which represented here as a scalar value $\bar{e}_t \in \mathcal{E}$ instead of the complete distribution E_t. Different characteristics of the distribution E_t can be used to obtain $\bar{e}_t$. Without loss of generality, we assume in the following the mean is used $\bar{e}_t = \mathbb{E}(E_t)$.

To be used as a decision policy, at each time epoch t_0 the optimal action $a_{t_0}^\star$ must be chosen based on current state (battery) and future expected energy production. To be fair in our comparison, we consider the same process is repeated at the beginning of each epoch, hence knowing the actual battery resulting of the realization of E_{t_0-1}.

Once uncertainties are removed, the problem is easily transformed as a linear program. We use notations for inputs and variables similar to the one used for describing the MDP. First, a non-linear obvious version can be written, using three decision variables per epoch. The accepted work at epoch t is denoted $a_t^\star$ and the corresponding amount of work done with this action, once the potential limited energy is taken into account, is denoted w_t^+. The third variable is the battery state b_t, with its initial value b_{t_0} set to either the state of the battery at the beginning of problem (when $t_0 = 1$) or to the resulting battery after the realization of the previous epoch decision.

$$\text{Maximize:} \quad \sum_{t=t_0}^{N-1} w_t^+ \cdot c^+ - (a_t^\star - w_t^+) \cdot c^- \tag{6.25}$$

subject to:

$$\forall t, \quad 0 \leq a_t^\star \leq w_t \tag{6.26}$$

$$w_t^+ = \min(a_t^\star, b_t + \bar{e}_t) \tag{6.27}$$

$$0 \leq b_t \leq B^{max} \tag{6.28}$$

$$b_{t+1} = b_t + \begin{cases} \eta^{-1} \cdot (\bar{e}_t - w_t^+) & \text{if } \bar{e}_t > w_t^+ \\ \bar{e}_t - w_t^+ & \text{otherwise} \end{cases} \tag{6.29}$$

$$\tag{6.30}$$

Several observations helps its linearization. First, it appears that the optimal accepted workload $a_t^\star$ is necessarily lower than the total available energy (battery and production), as higher values contributes negatively to the objective directly (by increasing the term $a_t^\star - w_t^+$) without helping in any indirect way. The two cases of b_{t+1} can be linearized in two inequalities. Its value should be as high as possible to maximize the objective, and is upper bounded both by $b_t + \eta^{-1} \cdot (\bar{e}_t - w_t^+)$ and by $b_t + \bar{e}_t - w_t^+$. Hence, the following linear program is equivalent:

$$\text{Maximize:} \quad \sum_{t=t_0}^{N-1} a_t^\star \cdot c^+ \tag{6.31}$$

subject to:

$$\forall t, \quad 0 \leq a_t^\star \leq w_t \tag{6.32}$$

$$a_t^\star \leq b_t + \bar{e}_t \tag{6.33}$$

$$0 \leq b_t \leq B^{max} \tag{6.34}$$

$$b_{t+1} \leq b_t + \eta^{-1} \cdot (\bar{e}_t - a_t^\star) \tag{6.35}$$

$$b_{t+1} \leq b_t + \bar{e}_t - a_t^\star \tag{6.36}$$

$$\tag{6.37}$$

As we chose a model with constant rewards c^+ over time, this linear program admit a trivial solution by choosing $a_t^\star = \min(w_t, b_t + \bar{e}_t), \forall t$. Doing so, the inefficiency of the battery is minimized and the total amount of work done is maximized. Interestingly, this leads to the same policy than the optimal for the Ideal-Bounded model, which was interpreted in remark 3, but for different reasons. This can be interpreted in this case as the limits of a deterministic production forecast. While the realization of energy is necessarily different from the prediction, taking this inaccurate forecast as the only known information prevent any accurate risk measurement. Hence, the optimal solution has to use as much of this energy as possible in order to avoid the inefficiency caused by the batteries. At the next decision epoch, if the realized energy production was higher than the prediction $\bar{e}_t$, the energy stored in the battery can be used at any time. The earlier is the better as it prevents either wasted energy due to full batteries and the inability to use all the stored energy because it is limited by the workload arrival near the end of the horizon.

Optimal policy with an oracle Our approach is based on an ε-optimal policy for maximizing the expected accumulated reward. It is however possible to compare it to an upper bound, which is represented by an optimal policy with perfect knowledge of future energy production. Such a policy may be built based on the detailed optimal policy with deterministic forecasts, using $\bar{e}_t = e_t$, where e_t is the realization of E_t as given by an oracle. In practice, this oracle is achieved by sampling the renewable energy distribution beforehand.

6.4.3 Methodology of experimental study

To attest the impact of uncertainty handling in this problem and of our approximation of the optimal policy, an experimental study is proposed using numerical simulations.

Implementation and practical considerations

The implemented approximation of optimal policy is based on the formulations proposed for error propagation analysis. A few adjustments are used for practical purpose, relying on existing implementations from SciPy [94] for some steps of the numerical computation. The core of the algorithm consists in computing values of $Q_t^{algo}(b, a)$ for different values of b and a. Two potentially costly computations are done once by epoch (t), reducing drastically the complexity and execution time of Q_t^{algo} without losing the error bound:

- the integral of $F_t(e)$ between 0 and any value up to $E^{\max}$ is computed with the desired maximum error $\frac{\varepsilon^{(1)}}{c^+ + c^-}$, allowing fast computation of r_t^{alg}
- the piecewise constant discretization of V_{t+1}^{algo}, discussed in section A.3, is computed such as it leads to a maximum error $\varepsilon^{(2)}$

The maximum of Q_t^{algo} for a given b is obtained using the SciPy `minimize_scalar` function with the `bounded` method, which is an implementation of Brent's algorithm. The maximum error of such algorithm is difficult to predict, depending on the shape of the function and of the number of iterations. Practical results show excellent accuracy and efficiency using this algorithm, even with a low number of iterations (≤ 50, converging usually in less than 20 steps). However its maximum error is not strictly bounded and we cannot give a value for $\varepsilon^{(3)}$ with this implementation but expect it to be negligible compared to the other errors.

For all the experiments presented thereafter, the other sources of errors are bounded by $\varepsilon^{(1)} = 0.01$ (error due to the reward approximation), $\varepsilon^{(2)} = 0.01$ (approximation of $P_t(b, a)$) and each linear interpolation of the value function is computed such as $\varepsilon^{(4)} < 0.1$. These values are absolute, their importance is therefore to compare with the other parameters used in the experiments such as the reward c^+, the maximum workload and the renewable energy production scaling.

Parameters of the model

The bounded, non-ideal batteries model is instantiated with different parameters along the detailed experiments. A summary of the default values, used when experiments are not conducted explicitly for studying different conditions, is given on table 6.1. As the considered model is continuous, the units can be set arbitrary as long as they are coherent with each others. To stay with order of magnitude which are easy to visualize and realistic for a fully renewable-powered datacenter, a relatively small infrastructure is proposed, with 20 kW peak solar panel production and $B^{\max} = 15\,\text{kW h}$ of usable battery capacity. All the experiments starts with a battery at half of its capacity. The datacenter peak

Symbol	Meaning	Value or distribution
$B^{\max}$	Usable battery capacity	15 kW h
$E^{\max}$	Installed peak capacity of photovoltaic array	20 kW
η	Battery efficiency	0.8 or 0.9
w_t	Workload arrival at time t	Uniform $\in [0, 40]$
c^+	Reward per unit of work done	1
c^-	Penalty per unit of undone accepted work	0.5
b_0	Initial battery capacity	$B^{\max}/2$

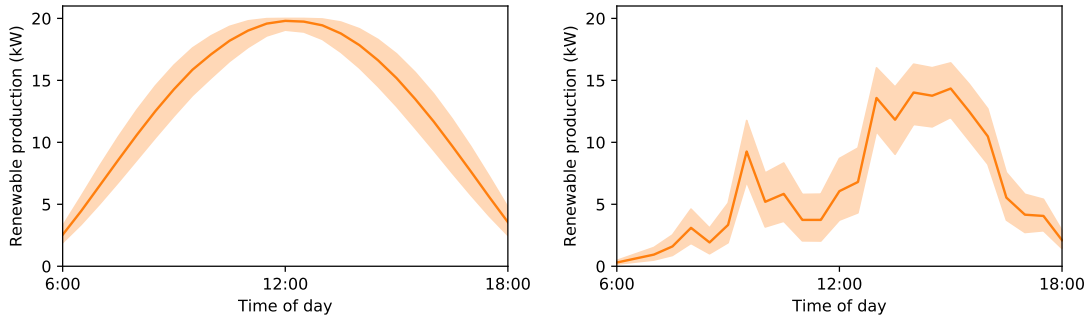
Table 6.1 – Summary of the model parameters used for experimental study (default values considered unless stated otherwise).

consumption is not explicitly represented in our model, but is obtained in the bounded workload formulation by the maximum workload arrival at each time step. Here, different values are used, representing a maximum capacity from 10 kW to 40 kW.

Choosing realistic values for the reward c^+ and penalty c^- is difficult. Existing datacenter operator and Cloud provider are usually accepting every submitted workload. They provide strong SLA guarantees but with low penalties in case of violations, often as *free credit* to use in the same service [5, 87]. The prospective datacenter model studied here behave quite differently, as a significant fraction of the work can be refused (without any cost). Hence, stronger reliability may be expected from the clients when some work is accepted, which prevent them from submitting it to another datacenter. We consider a unit value for the reward per unit of work done ($c^+ = 1$) and a base penalty of half ($c^- = 0.5$). For completeness, a study of the impact of penalty cost value is presented in the next section.

Battery efficiency varies highly with the technology used. In our model, additionally, this value represents the overall efficiency taking into account the required electrical converters, leading to slightly lower efficiency compared to the batteries alone. Two different values are considered, based on the efficiency range for different battery technologies of Dunn et al. [59]. Reasonable value for average lead-acid batteries is $\eta = 0.8$ and higher value of $\eta = 0.9$ models either average lithium-ion or very good lead-acid. While electrochemical batteries are usually in the 70% to 95% range, other energy storage devices (ESD) may be largely below (fuel cells combined with electrolyzer) or very close to a perfect efficiency (supercapacitors). Hence, an additional evaluation with other values $\eta = \{0.1, 0.2, \dots 1\}$ is performed to understand how different ESD would affect our approach.

Realistic PV production forecast data is relatively easy to find, as mentioned in section 3.1.2. This is unfortunately far more difficult when probabilistic forecasting is desired, as all the known datasets are limited to a single predicted value (deterministic models). Hence we used several models from the literature to build realistic setup of probabilistic PV forecasting. Deterministic irradiance and cloudiness data are extracted from NSRDB public dataset [169]. The variance of the prediction at each time is obtained through the empirical model of Kostylev and Pavlovski [106], which gives the error of best irradiance prediction method depending on prediction time ahead and cloudiness. Finally, this variance is used to build a Beta distribution with the parameters computed as described in [109], by using the original deterministic prediction as the mean of the distribution. The irradiance is finally converted to production forecast using an irradiance-linear model, sized such as its peak capacity is $E^{\max}$.



(a) Clear sky day

(b) Cloudy day

Figure 6.1 – Photovoltaic production forecasts with 95% probability intervals around the expected value.

We use NSRDB data from same location as experiments of chapter 5 (coordinates N34°85′ W120°45′, near the city of Santa Maria in California) but for the year 2017. Two different days are chosen for illustrating specific weather conditions, with their production for $E^{\max} = 20\,\text{kW h}$ shown in fig. 6.1. The clear sky day 17th July 2017 is an almost perfect day for PV production, with no significant clouds expected, keeping however some uncertainties due to all the other factors. Another day (7th May 2017) is used as a typical mostly cloudy day, causing relatively more uncertainties.

The experiments are made on a duration of 12 hours, with time steps of 30 minutes ($N = 25$ as the last epoch does not lead to any decision).

6.4.4 Methodology

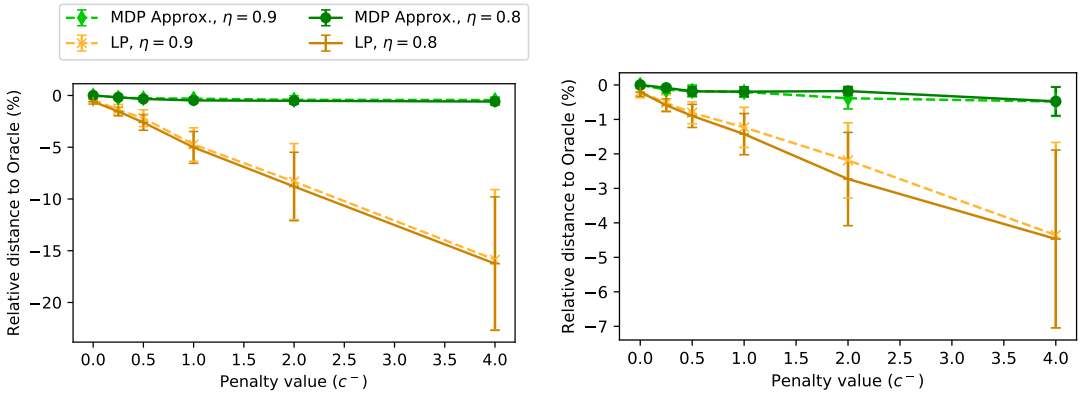
To evaluate accurately the different methods and compare their results, all are evaluated with the same random sampling of realized renewable production, according to the distribution E_t at each epoch t.

For a given experiment, the first step consists in solving the entire MDP using our ε-approximation algorithm, hence resulting in linear approximation of the V_t of each time step of the horizon. Then, each approach is evaluated across multiple runs, corresponding to different renewable productions during the time window. To do so, we start by sampling the realizations e_t of renewable energy for each $t \in \{1, \dots, N-1\}$, which are given to the oracle. One at a time, the approaches are used to take a decision for the first epoch, the realized energy is considered to compute the reward and the next battery and a new decision is taken according to the current battery state until the last epoch.

The workload arrival, while deterministic and perfectly known during the problem resolution, is sampled randomly from an uniform distribution. Each combination of parameters is evaluated with 5 different random workloads to mitigate the impact of favorable or unfavorable distribution. For every workload, the mentioned evaluation loop is repeated 100 times. The outcome of each approach is compared to the unquestionable result of the oracle-based optimization. This comparison is done for the same workload and renewable energy realizations, providing a robust metric to evaluate the quality of the results.

6.4.5 Simulation results

In a first series of experiment, the impact of different values of the failed work penalty (c^-) is evaluated. Figure 6.2 shows the results for the two different weather conditions and with



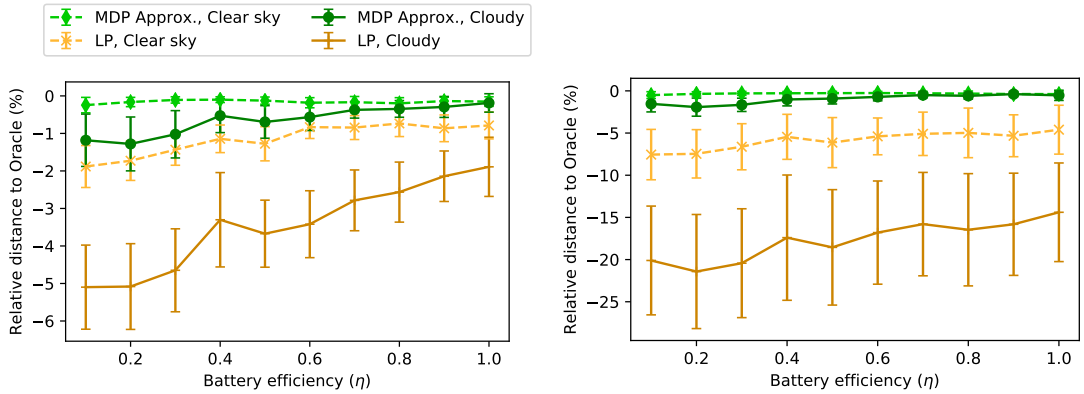
(a) With Cloudy production

(b) With Clear sky production

Figure 6.2 – Results for different values of penalty for accepted but not done work (c^-), while keeping the reward $c^+ = 1$.

battery efficiency η of either 80% or 90%. The values correspond to the relative differences of final accumulated reward over the whole horizon between oracle-optimal decisions and each approach, calculated for every identical conditions (same realization of renewable variables). The average over the 500 runs (100 per random workload) and the standard deviation are represented. With the cloudy day (fig. 6.2a), our approximation of optimal policy gives very stable performances, with a mean below 0.5% compared to the best possible outcome (oracle) and negligible standard deviation (< 0.3). At the contrary, the LP method with no uncertainty knowledge shows performances inversely proportional to penalty, from 0.6% when penalty is null to 16% for $c^- = 4$. It is also notoriously unstable, with its standard deviation growing similarly from 0.2% to 6.8%. For both approaches, the results are very close for both values of efficiency (less than 0.5% of absolute differences). The clear sky day, in fig. 6.2b, results in overly similar behaviors, with the LP method giving results closer to the oracle-optimal outcome (from 0.2% to 4.5%). While the scale of the figure shows better the slight decrease of our MDP approximation method, its mean performances are similar to those on Cloudy day, from almost perfect result ($< 0.001\%$) when penalty is 0 to 0.5% for $c^- = 4$. Its standard deviation for this high penalty is however slightly higher (0.43%). A minor difference is observed at $c^- = 2$ between the two values of battery efficiency, but being largely smaller than the standard deviation it may be attributed to some exceptional workload arrivals in several of the experiments.

In a second experiment series, a larger range of battery efficiency values η is used, from 10% to 100%. The performances of the approaches compared to perfect oracle are given in fig. 6.3a for the default, small penalty ($c^- = 0.5$). Globally, the results of the approaches are improving when getting closer to perfect batteries. For clear sky renewable production day, our method is between 0.24% and 0.1% from the oracle optimal outcome, with tiny deviation (≤ 0.2). The LP optimization with deterministic forecast gives quite good results in these conditions, with a mean of 1.8% for smallest efficiency and down to 0.8% for $\eta = 1$, but with higher deviation compared to our approach. In the cloudy day condition, both approaches performed significantly worst and the improvement caused by a better efficiency is more noticeable. The approximated MDP optimal policy is in average at 1.2% at lowest efficiencies (standard deviation of 0.7%) and reaches 0.19% (SD 0.24%) with perfect batteries. The LP method evolves similarly, starting at 5% (SD 1.1%) and reaching 1.9% (SD 0.8%). Both methods results in a noticeable irregularity when the



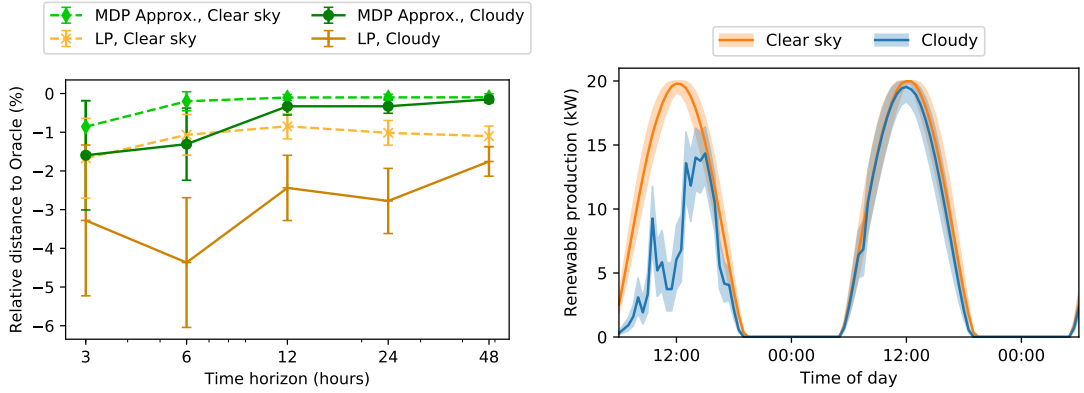
(a) Small penalty ($c^- = 0.5$)

(b) Large penalty ($c^- = 4$)

Figure 6.3 – Performance of the approaches compared to an oracle depending on the efficiency of the batteries (η).

efficiency is $\eta = 0.4$, performing slightly closer to the oracle compared to close values. While this could be due to favorable workloads scenarios during the evaluation process, it may also correspond to a non-linearity behavior of the optimal policies. For instance, this value may form a gap between cases where battery is unlikely to be full at any time (giving average production and workload) and cases where it becomes highly probable at some point.

Results of similar experiments with higher penalty ($c^- = 4$) are summarized by fig. 6.3b. Similarly to the observations on fig. 6.2, the LP method performs significantly worst with a large penalty. Its mean difference with optimal oracle final outcome varies from 7.5% to 4.6% with clear sky conditions and from 21% to 14% with the cloudy day. Compared to the smaller penalty, it also improves less with the efficiency of the batteries and is less stable (higher standard deviation in all tested conditions). Our approach behaves closer to the smaller penalty case, but also with slightly lower performances. The mean values varies from 1.9% for an efficiency of $\eta = 0.2$ to 0.5% for perfect batteries for the cloudy day and from 0.5% to 0.4% for clear sky day.



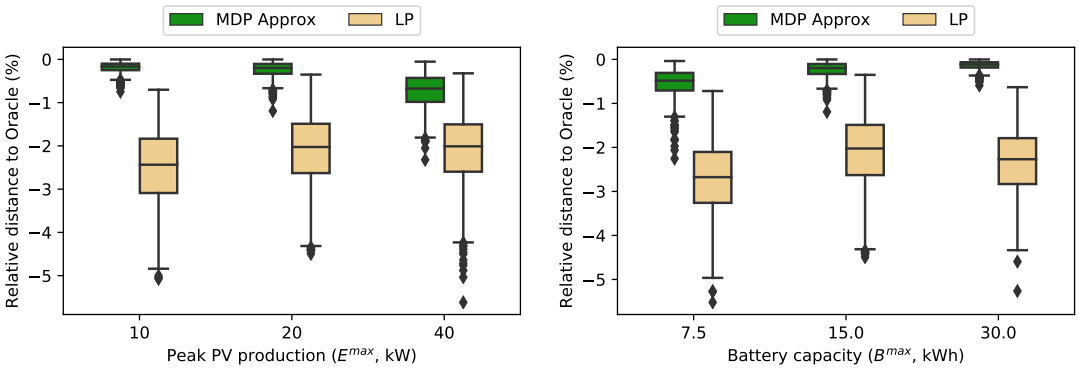
(a) Performances compared to oracle-optimal outcome (duration in log scale).

(b) Production over 48 hours for both weather conditions.

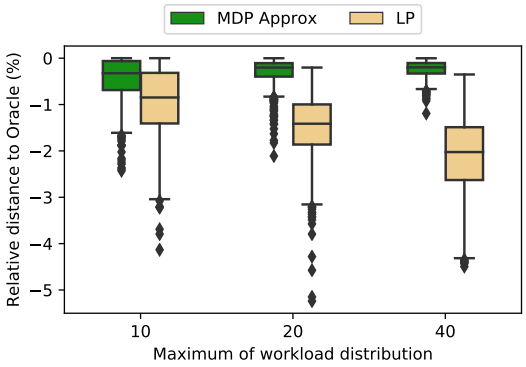
Figure 6.4 – Results for different time horizons, with number of epochs being 2 times the duration in hours (time step of 30 minutes) and efficiency $\eta = 0.8$.

The behavior of the approaches depending on the number of epochs is evaluated using fixed time steps of 30 minutes per epoch and adjusting the duration of the horizon between 3 hours and 48 hours. The extended power production with uncertainties for the two weather conditions is represented in fig. 6.4b. Despite his name, the *Cloudy* data is quite sunny in the second day. Figure 6.4a shows the results of LP and MDP approximation. Under both weather scenarios, our approach performs better with higher number of epochs and converges towards tiny difference compared to the oracle-based optimum (0.1% for clear sky and 0.15% for cloudy production). The LP method is always below the MDP approximation for similar weather. With the cloudy weather, it tends to give better results with longer time horizons, but with a few exceptions, regressing from a mean of 3.2% for a duration of 3 hours to 4.3% after 6 hours. In the clear sky case, the duration only slightly affects the results in a direction or another, with its worst value being for 3 hours (mean of 1.7%) and its best average results at 12 hours (0.8%). The observed convergence of both approaches between their results with clear sky or cloudy weather can largely be attributed to the second day, sunny in both conditions.

The other parameters of the model which can affect the results are finally studied. Figure 6.5 gives the results for each parameter using box plots, with the box showing the median and quartiles and the whiskers representing minimum and maximum excluding the (plotted) outliers.



(a) Impact of PV array sizing (peak production). (b) Impact of the battery usable capacity ($B^{\max}$).



(c) Maximum workload (upper bound of the uniform distribution of w_t at each t).

Figure 6.5 – Impact of parameters related to energy consumption and production, with battery efficiency set to $\eta = 0.8$ and the cloudy production day. Other parameters are set to their default values given in table 6.1.

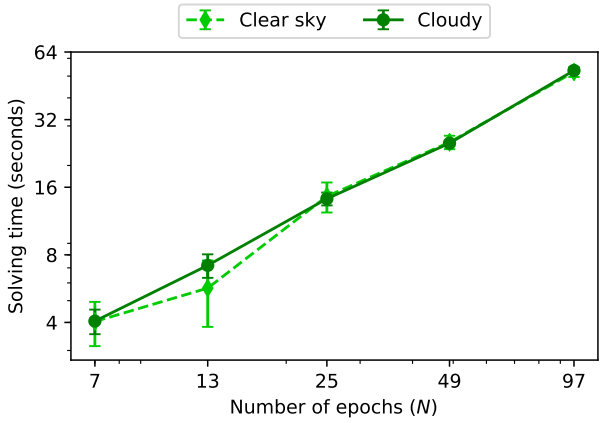


Figure 6.6 – MDP approximation solving time for different horizon size (log-log scale).

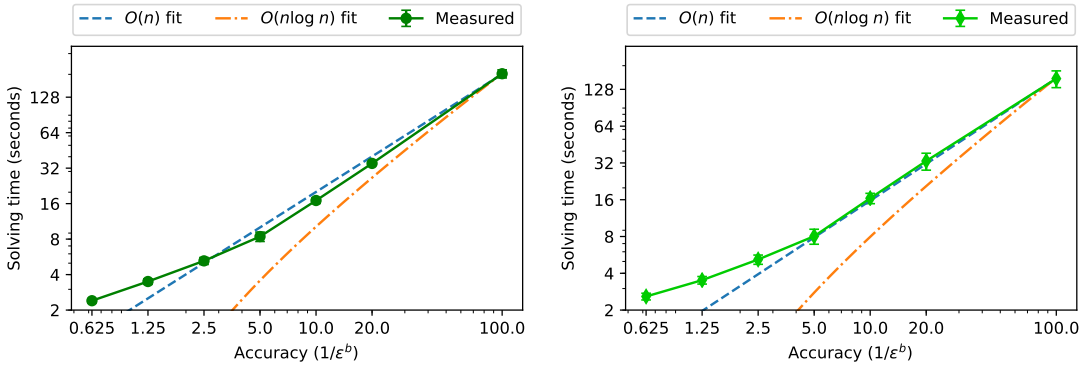
The effect of different sizing of the photovoltaic array, measured in terms of its peak production $E^{\max}$, is detailed in fig. 6.5a. The outcome of the LP method compared to the oracle is in the same range for all the values of $E^{\max}$, with median results between 2.5% and 2.1%. Our approach, while resulting in accumulated rewards closer to the absolute maximum, give slightly worst and more spread relative results with higher values of $E^{\max}$.

Figure 6.5b shows the impact of the battery capacity $B^{\max}$ on each approach. The LP approach with deterministic forecasts performs worst with small batteries (median difference with the oracle of 2.7%), followed by the larger setup (2.3%) and the medium capacity (2%). The MDP approximation improves its outcome with larger battery size in all points of view (quartiles and worst case), with a median value between 0.5% and 0.1%.

The workload at each time step is obtained from a uniform distribution between 0 and a maximum (40 in previous experiments). To measure the impact of different average loads, the maximum of this distribution is set to other values and the results of both approaches is given in fig. 6.5c. Without uncertainty distribution knowledge, the LP method gives worst relative results with more important workloads, with the median difference increasing from 0.84% to 2% while the dispersion stays similar (SD between 0.7% and 0.8%). Our method, at the contrary, both improves its median results and their dispersion with higher maximum workloads, from 0.3% (SD 0.5%) to 0.2% (SD 0.2%).

Execution time of the implemented approximation Along the discussions on error propagation and implementation, several remarks highlight ways to reduce the overall complexity and computation costs of our ε-approximation algorithm. However, we do not provide a formal analysis of its complexity. Some experimental results on the execution time are nevertheless provided here to give some indications of the actual cost of our implementation for solving the whole MDP with guaranteed maximum error, along with order of magnitude of the algorithmic complexity. Two important factors are expected impact significantly the solving time: the number of epochs in the horizon and the desired accuracy (maximal error tolerance per time step).

The number of epoch is increased by considering different horizon duration similarly to the results presented in fig. 6.4. Hence it varies between $N = 7$ (3 hours) and $N = 97$ (48 hours). The measured execution times, averaged over 5 runs, are presented in fig. 6.6 and used the default parameters (with $\eta = 0.9$). Both renewable production scenarios result in almost identical execution times, at the exception of a data point at $N = 13$ where clear sky weather is slightly below but within the standard deviation range. The execution time



(a) For the cloudy scenario.

(b) For the clear sky scenario.

Figure 6.7 – Solving time of MDP approximation depending on the desired maximum error, with corresponding $n \to an$ and $n \to an\log(n)$ best fits, where n is the accuracy (log-log scale).

of our algorithm is close to perfectly linear with the number of epochs solved.

Measuring how the maximum desired error of the approximation affects the solving time requires first to define a measurement of the accuracy and the distribution of the errors across the different sources of errors. We denote here ε^b the base maximum error, which is used to defines the values of each error source bound such as $\varepsilon^{(1)} = \varepsilon^{(2)} = 0.1 \cdot \varepsilon^b$ and $\varepsilon^{(4)} = \varepsilon^b$. As mentioned earlier, our implementation does not strictly guarantee a maximum error on $\varepsilon^{(3)}$, but with reasonable number of iterations of the maximum finding algorithm it appears to be negligible compared to the others. Hence the maximum total absolute error for each time step is given by $\varepsilon_t = 1.2\varepsilon^b$. To make the results easier to interpret, we define *accuracy* as the inverse of ε^b. Hence, dividing the error by x is equivalent to the more intuitive representation of an accuracy improved by a factor x.

Measured execution times, with default parameters (and $\eta = 0.9$) are given for each weather condition in fig. 6.7. A difference is measured for the highest accuracy between the two scenarios, with a mean of 157 seconds (SD 24.6) for clear sky, compared to the 202 seconds (SD 15.9). The other measurements are all far closer from each other than the standard deviation. For each production scenario, both $n \to a \cdot n$ and $n \to a \cdot n\log(n)$ functions are shown, using least square method for fitting the value of a and n being the accuracy. The solving time with clear sky condition (fig. 6.7b) is relatively well approximated by a linear function of accuracy. Asymptotically, it seems to increases very slightly faster, but significantly below an $n\log(n)$ function. For cloudy weather, represented in fig. 6.7a, the overall execution time appears clearly to be superlinear with the accuracy, as shown by the linear fit, but is also growing slower than the $n\log(n)$ function.

6.4.6 Discussion

In the experiment results, two methods were primarily compared with each other: our bounded-error approximation of optimal MDP policy, taking uncertainties fully into account and the repeated linear programming method, providing the optimal decision under deterministic forecasts assumption. By measuring their performances in terms of relative difference compared to the absolute best, oracle-based decisions, two major pieces of information are available. The first is given by the results of our approximation alone, as a measurement of how the uncertainties are intrinsically affecting the outcome even in case

of a policy knowing their distribution (but obviously not their future realizations). The second, provided by the comparison between the MDP approximation and the LP results, indicates the potential improvement that may be obtained by fully considering the random distribution of the variables instead of a single point prediction (the average of the distribution). Such a comparison is useful in our context, as renewable production forecasts are still largely modeled as deterministic predictions in existing datacenter management approaches.

In the problem detailed along this chapter, three kind of penalties may affect the reward obtained in case of unexpected renewable production. If more power is produced then expected, the battery efficiency causes a loss of energy compared to directly using it and consequently a lower final reward even if the battery energy is used before the end of the horizon. In case the battery is not able to store all this exceeding energy, additional losses are caused by the battery becoming full. At the contrary, if less energy is produced than expected, the penalty c^- may reduce the total gain. Except for the battery full losses, which is not likely to happen in practice, these penalties are added linearly with the amount of the prediction error.

Linear penalties are usually causing less trouble for approaches which ignore the uncertainties, especially when the distributions of random variables are narrow. This is shown by the results of LP which give a total outcome closer than 6% of the oracle-optimal under many tested conditions, when the work failed penalty is small ($c^- \leq 0.5$). The relative results are even better when the renewable production is distributed closely around the expected value, as attested by the LP outcome being in average closer than 2% for similar conditions when clear sky weather scenario is used. Such a deterministic optimization leads, however, to quite unstable results which cause high standard deviation and distribution of the relative total reward across multiple runs differing only by the realization of the renewable energy production. In addition, considering higher penalty for failed work, such as $c^- = 4$ affects considerably the average outcome of this method, which is unable to estimate the risk associated to a decision.

By using a MDP formulation, hence taking fully into account the notions of risks and random variable distributions, our bounded approximation algorithm performs considerably better and more stably in all the tested conditions. Notably, the average relative result compared to the oracle-based optimization is always less than 2%, despite some experiments with unfavorable parameters (such as a low number of epochs, high failed work penalty and bad battery efficiency). This number is even lower when considering the more stable, clear sky forecast ($< 1\%$ from the oracle). The different experiments show the robustness of our approach for different setup of model parameters corresponding to realistic or even hypothetical infrastructures setups. While its impact is not null, the MDP handles very well cases with low or high penalty c^- for work accepted but not done.

The performances compared to an oracle are the most affected negatively by low battery efficiency ($< 40\%$) and short time horizon ($<$ 12 hours or 25 epochs). The first is explained by the reduced ability to store excess of energy for latter use, preventing the optimal policy from relying on this energy buffer to compensate the unavoidable uncertainty of renewable energy. Short time horizons is impacting the relative performances mostly because our objective ignore the final energy in batteries (after the last decision epoch). The oracle, knowing the last realization of energy, is assured to end with an empty battery if it is possible. At the contrary, the MDP can only takes the optimal risk which gives the higher expected reward for the last energy distribution. As the final outcome is the accumulated rewards along the whole horizon, the relative impact of this last step is more important with short horizons and become negligible with longer ones.

The impact of other parameters on the MDP approximation algorithm results is limited, but some correlations appear. Larger renewable energy production may, counter-intuitively, reduce the performances of our approach compared to the oracle-based optimization. This observation is explained by the higher probability of exceeding the battery capacity at least once during the horizon in this case, similarly to the more intuitive effect caused by lowering the battery capacity.

An empirical study of the complexity of our approximation is provided by the measurements of execution times under several conditions. Because the outcome at a given epoch depends only on the instant reward and on the values of each state of the next epoch, even generic solving algorithms have rarely more than a linear component of their complexity depending on the number of epochs. Our algorithm, as expected, is requiring a computing time linear with the total epochs of the horizon. This computing time is slightly superlinear with the desired accuracy (inverse of the maximum error), but growing slowly than $n \log(n)$ functions. Traditional and generic methods to approximate MDP with continuous states and actions rely on a discretization of both, which can also achieve arbitrary accuracy in theory by using enough discrete values. Such discrete problem with S different states and A actions can be solved by dynamic programming, with a complexity of $O(S^2 \cdot A)$ for each epoch. Doubling the accuracy, without specific structural properties, requires to double the number of discrete states and actions. Hence, a simple discretization would cause the computing time to depends on the cube of the desired accuracy. Our approach, while giving a proven ε-approximation of the optimal policy, scales exceptionally well with the desired accuracy. This is made possible by exploiting the analyzed structure of the optimal policy, which gives a more efficient formulations, along with the detailed procedure for discretizing the continuous functions which reduces computations while guaranteeing a known maximum error.

6.5 Conclusion

Along this chapter, we presented a MDP formulation of optimal workload acceptance problem in datacenter powered by uncertain RES and ESD. By studying the properties of the MDP, we show that optimal policies have several remarkable properties. This structure is usable to reduce the complexity of finding a bounded approximation of an optimal policy compared to arbitrary continuous state and action MDP.

Such an efficient algorithm has been implemented, with a detailed analysis of the sources of errors and methods to bound them. Hence, the worst case error of our approximation compared to the uncomputable, theoretical optimum is known and can be chosen arbitrarily small. This ε-optimal approach has been evaluated by numerical simulations, using realistic renewable production forecasts. It has been compared both to the optimal policy considering deterministic (yet inexact) weather forecasting and to the best achievable results using an oracle to predict the renewable energy production. These results highlight the stability and accuracy of our approach, which perform closely to the oracle-based optimization (less than 2% of average difference in all tested conditions). By fully taking random distribution of uncertain variables into account, our approach improves significantly the expected total gain during the optimization horizon, as well as the worst case and a lower variability of the results.

This mean improvement of the total gain depends on many parameters of the model, with the two most important criteria being the uncertainties of the weather conditions and the failed work penalty. The model studied allows the datacenter to refuse, without any cost, a part of the workload before the beginning of a decision epoch. Therefore, a strong

guarantee that accepted works will be executed must be given to the users, modeled as a penalty to pay back for any work that failed to be executed in addition to not receiving the work reward. Contrary to a method using deterministic forecasts, our approach handles well important values for this penalties, with up to 20% improvement of the total gain for a penalty 4 times higher than the corresponding reward. Such an important penalty may appear excessive, but shows how the knowledge of RES uncertainties can be used to provide reliable service, with important guarantees for the users, without reducing significantly the total benefit.

The implementation of the MDP approximation algorithm has been shown experimentally to be efficient in terms of computing time depending on the number of epochs and on the maximum approximation error, making it usable in practice even when high accuracy is required. As average execution times on a few scenarios can hide the worst cases, additional analysis is nonetheless necessary to find its algorithmic complexity, giving some guarantees on the solving performances. This is therefore an important extension of this work to consider in the near future.

By providing a comprehensive analysis of the problem through MDP formulation along with an efficient solving method, this work opens several perspectives. A first direction to explore is the extension of the underlying model, to take into account more of the complexity of real-world infrastructures. Based on our analysis, some extensions can be added with less extensive proofs. For instance, by showing that a deterministic but time-dependent reward per unit of work does not impact the different hypothesis, it is possible to reuse most of the proofs with few or no modifications. Our approach can also be used as a base to target an infinite-horizon variant of the problem, for instance by using myopic methods or looking for stationary policies. Some extensions are however expected to be difficult to integrate in this analytical framework. Particularly, if many pieces of information must be maintain between decision epochs, the Markov property requires them to be part of the MDP state. At the same time, MDP with large state space are both very costly to solve exactly and more difficult to study in order to find structural properties.

Finally, we studied here a single source of uncertainty which is the renewable energy production. When optimizing a real-world infrastructures, uncertainties are present in almost every aspect, either caused by imperfect models of deterministic processes (execution time of batch jobs or resources consumption of a service for a given request) or by intrinsically unpredictable events (task submission rate from users or hardware failures for instance). Considering the datacenter model used in this chapter, the workload arrival is supposed to be known perfectly for the whole horizon. It could be modeled instead as a stochastic process to take into account uncertainties in both energy production and consumption.

Chapter 7

Conclusion

An unexciting truth may be eclipsed by a thrilling falsehood.
— Aldous Huxley, *Brave New World Revisited*

With the rise of information and communication technologies, the number and the size of datacenters is rapidly growing nowadays. To mitigate the ecological impact caused by their huge energy consumption, on-site renewable energy sources (RES) combined with some energy storage is a promising perspective. It also becomes more and more economically attractive with the simultaneous rise of energy cost and fall of some RES technology prices such as photovoltaic panels. Leveraging solar and wind energy in a datacenter is nonetheless challenging, as they are intrinsically intermittent and uncertain, whereas conventional datacenter management relies on constant and reliable power supply. This thesis investigated how these two key characteristics of solar and wind energy can be integrated in datacenter management methods in order to use RES as efficiently as possible.

Several approaches have been presented, each work exploring a different aspect of the overall problem. First, a scheduling algorithm designed to deal with intermittent sources and variable cost of energy was proposed. Contrary to existing renewable-aware approaches, the electrical model and primary data, such as the prediction of renewable energy production, are not directly required by the algorithm. Limited knowledge is used instead, by the way of an independent system managing the electrical infrastructure and giving preference indications to the scheduler depending on its internal objectives. The interest of this black-box model was demonstrated by evaluating two variants of the electrical management, without any changes required in the scheduling algorithm itself. Both variants reached similar performances compared to an improved version of GreenSlot, an algorithm from the literature solving the problem by using a centralized view with both IT and electrical models and data.

The second contribution explored the idea of black-boxes further, by considering independent and opaque management of electrical and computing infrastructures. Using multi-objective evolutionary algorithms (MOEA) and fitness approximation techniques, it solves the overall power planning problem in a purely multi-objective way and without depending on specific electrical or computing models.

These two contributions show the feasibility of cooperative optimization of independent management systems, one taking care of machines and tasks, the other optimizing the electrical sources and energy storage devices (ESD). Such a co-optimization is expected to be less efficient than a global and centralized management with full knowledge of both parts. In practice however, a perfect management of the overall system is hard to achieve. This is mainly due to of the diversity of possible setups for both the data-

center and its on-site RES and ESD, along with the immense complexity of fine models of both infrastructures. Being able to design independently the electrical and computing optimization systems increases the flexibility (by uncoupling the two infrastructures) and makes it easier to finely optimize each part (by separating the concerns, hence reducing the complexity of the considered models and allowing researchers of each domain to focus on their area of expertise).

The last approach differed from the others by studying the holistic problem in a centralized way. Contrary to the intermittency, uncertainties of RES powering datacenters have been studied in very few works. A more theoretical study has been proposed, dropping the black-box abstraction to provide a better understanding of how uncertainties impact the problem. By modeling it as a Markov Decision Process (MDP), the structure of the optimal policy has been deeply analyzed. This structure has been leveraged to design an algorithm for taking decision with proven bound from optimum, yet reducing sufficiently the complexity to be usable on large instances. The uncertainties are rarely considered in existing works, either totally ignored by using perfect predictions or their effects slightly mitigated by handling incorrect predictions once they are detected. While the datacenter and workload model is arguably simplified, this work provides robust foundations for tackling more realistic models. Our simulation results have shown the benefits of accounting uncertainties as accurately as possible, in the core of the datacenter management methods, increasing the robustness and reliability of the decisions.

In addition to the approaches proposed to tackle intermittency and uncertainty aspects of RES in datacenter management, works on simulation and validation tools have been detailed. By studying the integration of electrical infrastructure models in general-purpose datacenter simulators, a contribution to the methodological tools has been proposed. Moreover, a large-scale datacenter workload based on Google cluster traces has been studied and its statistical characteristics analyzed. This work has led to a synthetic workload generator, with reduced and understandable set of parameters, allowing more robust and reproducible experiments.

7.1 Perspectives

The contributions presented in this manuscript raise some new research questions and many perspectives may be envisioned, both as short-term research projects and broader, long-term directions.

7.1.1 Short-term perspectives

Power consumption from non-computing parts

All the contributions presented here considered only the computing power consumption. In a datacenter, other devices are consuming significant amount of power, with the most important being generally networking devices and cooling systems. Several extensions of the proposed approaches may be considered to take their power needs into account. A first possibility is to simply model them as part of the overall IT infrastructure. Simplistic modeling of cooling consumption may be added through Power Usage Effectiveness (PUE) factor, for instance. More advanced approaches may be used to allow the cooling system to leverage its temporal flexibility, *i.e.* by cooling slightly more than necessary when renewable energy is available to reduce the consumption afterwards [120].

The second possibility is to consider each other system as a partially-independent black-box. As for the cooling system, it needs to interact both with the IT task and

machine management (to provide the required cooling) and with electrical infrastructure (to take energy availability into account).

Co-simulation of datacenters and electrical sources

As of today, it is difficult for researchers working on RES integration to datacenters to have an access to real testbed. Hence the design and validation of approaches is usually performed by simulation. However, no single generic simulation platform emerged in the community, leading to many works being evaluated using ad-hoc numerical simulation. In addition to taking precious time to researchers in software development and debugging, it makes more difficult to reproduce approaches and their results.

The work on DCworms presented in section 3.1 showed it is possible to extend existing datacenter simulators, which are already heavily used and supported by the energy-aware datacenter management community, to support rich electrical infrastructure simulation. Unfortunately, this simulator extension has been poorly diffused and it was difficult to merge the modifications in the main codebase maintained by the PSNC laboratory. To be really useful, such an integration must be similarly supported by a significant part of the renewable-aware community. This requires primarily to identify the most important needs, which is a difficult task when considering the variety of existing works in terms of used models. This variety is mainly explained by the different trade-off choices between accuracy and ease of use of the models, but also by the youth of the research area.

Therefore, a common effort from different actors of the community seems critical to design, develop and support a comprehensive simulation platform. While the initial cost to invest is currently perceived as too high for each researcher or team, the overall benefit seems to justify the effort. We argue that such simulation platform is the next important step in the maturation of the renewable-aware datacenter management area. Such a common platform is critical for the quality of future research, for making it easier to setup realistic models and reducing the current burden of re-implementing the simulation process used by existing works for comparing with them.

Many objectives and sequential decision making

A large proportion of the existing works tackles the renewable-powered datacenter management as a single objective problem, after considering either the quality of service or some environmental criteria as a hard constraint. We argued in the manuscript the intrinsically multi-objective aspect of the overall problem: even if the service level agreement incorporates economical compensation in case of QoS violation, this should be considered in the trade-off with ecological impact. The contribution detailed in chapter 5 showed the feasibility of multi-objective power planning optimization resulting in a comprehensive set of Pareto-optimal trade-offs. However, the choice of a specific solution has not been explored and is let to an abstract decision maker.

When multiple decision epochs are considered, the overall problem becomes a multi-objective sequential decision making problem. Future works may explore this long-term variant, for instance through multi-objective reinforcement learning [189] or using multi-objective Monte-Carlo tree search [196]. It is worth noting that these approaches require, in addition to the reward on each objective associated with a possible decision, information on the current state of the overall system, *e.g.* the state of charge of the various ESD or the waiting and running workload. Oppositely, black-boxes as formulated in chapter 5 are designed to prevent state-related information to be accessible. Hence additional study should explore how some aspects of the internal IT and electrical states could be exposed

while preserving as much as possible the benefits of a black-box design. Formulated differently, it means finding minimal representation of internal states for these "gray-boxes" to provide enough information for inferring the impact of a decision (implementing a possible power profile) on their future capabilities and constraints.

Another perspective is the extension to more objectives. The cooperative multi-objective optimization presented here used two objectives, one per decision module: ecological impact and quality of service. The approach may be extended to handle more objectives, some of them being shared between decision modules. For instance, the economical aspect of a public datacenter depends both on energy cost and revenue from users submitting the workload. Taking the economical objective into account, without ignoring the others, is a promising perspective to improve the acceptance of such approach by existing datacenter operators.

Other sources of uncertainty

Renewable source production is certainly an important, but not the only source of uncertainty. The workload of the datacenter contains many uncertain dimensions, even outside the context of renewable-aware management. First, whatever the workload type is (batch-like, services or mix of them), its arrival is rarely known with precision. Similarly, the resource consumption of a given task or of a given service is seldom known with certainty, especially if the users that submit the workload are exteriors from the datacenter operator (public and hybrid clouds for instance). Finally, for batch workload specifically, the duration of a task is usually estimated by the user, but with a large margin of error.

As we showed in section 3.2, the different aspects of a given workload obey to some distribution laws. In some contexts, such as interactive services for end users in a geographically limited area, cyclic patterns due to time of activity (daily work hours, week-end...) can also be used to predict expected future load. By taking into account both prediction and its uncertainty, the computing resources and the ESD may be used more efficiently by anticipating the most likely scenarios as well as some less probable ones. The method used to study the problem in chapter 6 is not easily extendable to some complex models, specifically if they require to maintain many state information as high dimensional states MDP are especially hard to analyze. Large states may be need for instance to keep track of long tasks requiring several decision epochs to be completed. Other methods can be used to increase the robustness of existing approaches while increasing the outcome in most probable scenarios, such as Monte Carlo simulations techniques and stochastic programming.

7.1.2 Long-term directions

Towards cooperative datacenters on the smart grid

Along this thesis, we focused on the management of a single datacenter with its own micro-grid including RES and ESD. Many works show the benefits of considering multiple, geo-distributed datacenters cooperating or competing when RES are used. Doing so adds a new mechanism to handle intermittent production, which is the spatial distribution of weather conditions. It also provides better models of datacenter integration in a large (global or national) electrical grid: these energy-hungry facilities are promising candidates for demand-response scheme designed by the smart grid community.

Nowadays, works are almost hermetically partitioned between single datacenter with on-site renewable production and geo-distributed datacenters management. In the former setting, the research focuses usually on realistic computing resources and workload models,

at the cost of simplified power production and storage devices representation, often without modeling the complex behaviors of a smart grid. Oppositely, the latter geo-distributed works rely generally on more realistic models of electrical sources and grid behaviors, but ignore many important aspects of the workload and of the machines.

In a near future, with the rise of distributed small and medium-scaled RES power plants and the modernization of regional electrical infrastructure, the smart grid will likely become a more and more tangible concept. The IT area is also evolving towards more distributed vision instead of huge centralized datacenters, as shown by the recent interest in concepts such as the *Edge Computing* [173]. In this context, a promising research direction is a closer integration of datacenter management with the smart grid. The existing geo-distributed works give many clues on the challenges and opportunities of this model. Future works should focus on using more realistic demand-response schemes together with detailed IT models and study cooperative mechanisms to distribute the workload among distributed datacenters.

Questioning the humans behind the machines: users and developers

Current researches on sustainable datacenters usually assume an implicit hypothesis: compared to current usage, the same service is expected from users, with few or no modifications of existing software providing it. Hence, few works study how end-users accept potential lower service quality if they are helping RES integration. Similarly, taking into account intermittency and uncertainty of RES at the application level is rarely studied, with the notable exception of embedded systems area [83].

Meanwhile, the ecological awareness is growing, with more and more people understanding (or at least trying to understand) the implications of their actions on the environment and accepting some trade-offs between their desire and the ecological consequences. To avoid the increased energy efficiency of datacenters from leading into a greater overall energy consumption due to the rebound effect [69], the behaviors of users and developers must be taken into account. Future works should investigate the trade-offs between performances and environmental impact accepted by users in the context of computing services, which is a central aspect of a global sustainability. Some real-world, empirical experiments may be pointed out, such as the website of *Low-tech Magazine*[1] running on a small server powered only by photovoltaic panels and a battery. Instead of sizing the electrical infrastructure for the needs of the existing, always-available website, they redesigned it entirely to minimize its power need (using low-resolution illustrations and less modern-looking style for instance) and accept occasional disruptions when several days of unfavorable weathers occur in a row.

Two complementary axes appear therefore interesting. The first is to give more ways for the end users to adapt their trade-off between quality or reliability and environmental impact. This requires both to understand the expectations of these users, which involves sociological studies, and providing enough information to help them to do a choice with all required knowledge. The second axis is the modification of the software, from the operating systems to the development frameworks and applications themselves, in order to make them more sustainable and to take into account user preferred trade-offs. Making applications more sustainable may mean first taking energy consumption into account during development phases. But in the context of interactive services running on machines powered by intermittent RES, the applications could also be designed to integrate well with a variable energy availability: supporting degraded, low-consumption modes or handling

[1] Low←tech Magazine website: https://solar.lowtechmagazine.com/about.html

temporary failures in some cases.

Obviously, some of the datacenter use-cases are more difficult to adapt in this way than others. However, these possibilities are still largely unexplored as of today, at a time where a sustainability without significant changes of behaviors seems more difficult to envision than ever.

Publications

International journals

- Jean-Marc Pierson, Gwilherm Baudic, Stéphane Caux, Berk Celik, Georges Da Costa, Léo Grange, Marwa Haddad, Jérôme Lecuivre, Jean-Marc Nicod, Laurent Philippe, Veronika Rehn-Sonigo, Robin Roche, Gustavo Rostirolla, Amal Sayah, Patricia Stolf, Minh-Thuyen Thi, and Christophe Varnier. DATAZERO: Datacenter With Zero Emission and Robust Management Using Renewable Energy. *IEEE Access*, 7: 103209–103230, 2019. doi: 10.1109/ACCESS.2019.2930368
- Léo Grange, Georges Da Costa, and Patricia Stolf. Green IT scheduling for data center powered with renewable energy. *Future Generation Computer Systems*, 86: 99–120, September 2018. ISSN 0167-739X
- Georges Da Costa, Léo Grange, and Inès de Courchelle. Modeling, classifying and generating large-scale Google-like workload. *Sustainable Computing: Informatics and Systems*, 19:305–314, September 2018. ISSN 2210-5379. doi: 10.1016/j.suscom.2017.12.004

International conferences and workshops

- Georges Da Costa, Leo Grange, and Ines De Courchelle. Modeling and generating large-scale Google-like workload. In *2016 Seventh International Green and Sustainable Computing Conference (IGSC)*, pages 1–7, Hangzhou, China, 2016. IEEE. ISBN 978-1-5090-5117-5. doi: 10.1109/IGCC.2016.7892623

National conferences and workshops

- Léo Grange, Patricia Stolf, Georges Da Costa, and Paul Renaud-Goud. Négociation multiobjectif de profils de puissance de centre de données alimenté par énergies renouvelables sur site. In *Conférence d'informatique En Parallélisme, Architecture et Système*, Toulouse, France, July 2018
- Léo Grange, Patricia Stolf, Georges Da Costa, and Amal Sayah. Heuristiques d'ordonnancement pour les centres de données alimentés par énergies renouvelables. In *Conférence d'informatique En Parallélisme, Architecture et Système*, Sophia Antipolis, France, June 2017

Under review

- Léo Grange, Patricia Stolf, Georges Da Costa, and Paul Renaud-Goud. Cooperative and Multi-Objective Power Planning Negotiation between Datacenter and on-Site Renewable Energy Sources. In *Submitted to IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2019

– Gustavo Rostirolla, Léo Grange, Minh-Thuyen Thi, Patricia Stolf, Jean-Marc Pierson, Georges Da Costa, Gwilherm Baudic, Marwa Haddad, Ayham Kassab, Jean-Marc Nicod, Laurent Philippe, Veronika Rehn-Sonigo, Robin Roche, Berk Celik, Stéphane Caux, and Jérôme Lecuivre. Sizing and Management of Energy Sources for Green Datacenters with Renewable Energy. *submitted to Elsevier Renewable Energy*, 2019

Others

– Léo Grange, Patricia Stolf, Georges Da Costa, and Paul Renaud-Goud. Cooperative and multi-objective power planning negotiation between datacenter and on-site renewable energy sources. Research Report IRIT/RR–2018–07–FR, IRIT, December 2018

Appendices

Appendix A

Full proofs of Chapter 6: imperfect batteries

A.1 Unbounded batteries and workload

A.1.1 Partial differentiates $Q_t(b,a)$

Case $a \geq b$ We recall that if $a \geq b$, $Q_t(b,a) = r_t(b,a) + \sum_{i=1}^{3} Q_{t,3}(b,a)$, and we already computed

$$\frac{\partial Q_{t,1}}{\partial b}(b,a) = \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b+\eta e)de$$
$$\frac{\partial Q_{t,2}}{\partial b}(b,a) = -V_{t+1}(0) \cdot q_t(a-b).$$

$$\begin{aligned}
\frac{\partial Q_{t,3}}{\partial b}(b,a) &= \int_0^1 q_t(a-be) \cdot V_{t+1}(b(1-e))de \\
&\quad + b\Big(\int_0^1 \Big((-e) \cdot q'_t(a-be) \cdot V_{t+1}(b(1-e)) \\
&\qquad + (1-e) \cdot q_t(a-be) \cdot V'_{t+1}(b(1-e)) \Big) \Big) de \\
\frac{\partial Q_{t,3}}{\partial b}(b,a) &= \int_0^1 q_t(a-be) \cdot V_{t+1}(b(1-e))de + b\int_0^1 q_t(a-be) \cdot V'_{t+1}(b(1-e))de \\
&\quad + \int_0^1 \Big(e \cdot (-b) q'_t(a-be) \cdot V_{t+1}(b(1-e)) \\
&\qquad + e \cdot q_t(a-be) \cdot (-b) V'_{t+1}(b(1-e)) \Big) de
\end{aligned}$$

We integrate by parts the last integral, differentiating $e \mapsto e$ and integrating $e \mapsto (-b)q'_t(a-be) \cdot V_{t+1}(b(1-e)) + q_t(a-be) \cdot (-b)V'_{t+1}(b(1-e))$

$$\begin{aligned}
\frac{\partial Q_{t,3}}{\partial b}(b,a) &= \int_0^1 q_t(a-be) \cdot V_{t+1}(b(1-e))de + b\int_0^1 q_t(a-be) \cdot V'_{t+1}(b(1-e))de \\
&\quad + [e \cdot q_t(a-be) \cdot V_{t+1}(b(1-e))]_0^1 - \int_0^1 q_t(a-be) \cdot V_{t+1}(b(1-e))de \\
\frac{\partial Q_{t,3}}{\partial b}(b,a) &= b\int_0^1 q_t(a-be) \cdot V'_{t+1}(b(1-e))de + q_t(a-b) \cdot V_{t+1}(0)
\end{aligned}$$

Altogether

$$\begin{aligned}\frac{\partial Q_t}{\partial b}(b,a) &= (c^+ + c^-)\cdot F_t(a-b) + \int_0^\infty q_t(a+e)\cdot V'_{t+1}(b+\eta e)de \\ &\quad - V_{t+1}(0)\cdot q_t(a-b) + b\int_0^1 q_t(a-be)\cdot V'_{t+1}(b(1-e))de \\ &\quad + q_t(a-b)\cdot V_{t+1}(0) \\ \frac{\partial Q_t}{\partial b}(b,a) &= (c^+ + c^-)\cdot F_t(a-b) + \int_0^\infty q_t(a+e)\cdot V'_{t+1}(b+\eta e)de \\ &\quad + b\int_0^1 q_t(a-be)\cdot V'_{t+1}(b(1-e))de\end{aligned}$$

We obtain $\frac{\partial Q_t}{\partial a}(b,a)$ similarly.

$$\begin{aligned}\frac{\partial r_t}{\partial a}(b,a) = c^+ - (c^+ + c^-)\cdot\Big(&\int_0^1 F_t((a-b)e)de \\ &+ (a-b)\cdot\int_0^1 eq_t((a-b)e)de\Big)\end{aligned}$$

Then we integrate by parts the second integral, differentiating $e \mapsto e$ and integrating $e \mapsto (a-b)q_t((a-b)e)$

$$\begin{aligned}\frac{\partial r_t}{\partial a}(b,a) &= c^+ - (c^+ + c^-)\cdot\Big(\int_0^1 F_t((a-b)e)de \\ &\qquad + [eF_t((a-b)e)]_0^1 - \int_0^1 F_t((a-b)e)de\Big) \\ \frac{\partial r_t}{\partial a}(b,a) &= c^+ - (c^+ + c^-)\cdot F_t(a-b)\end{aligned}$$

$$\frac{\partial Q_{t,1}}{\partial a}(b,a) = \int_0^\infty q'_t(a+e)\cdot V_{t+1}(b+\eta e)de$$

We get rid of the q'_t to keep only expressions where q_t is not differentiated. To do that, we integrate by parts, integrating $q'_t(a+e)$ and differentiating $V_{t+1}(b+\eta e)$.

$$\begin{aligned}\frac{\partial Q_{t,1}}{\partial a}(b,a) &= [q_t(a+e)\cdot V_{t+1}(b+\eta e)]_0^\infty - \eta\int_0^\infty q_t(a+e)\cdot V'_{t+1}(b+\eta e)de \\ \frac{\partial Q_{t,1}}{\partial a}(b,a) &= -q_t(a)\cdot V_{t+1}(b) - \eta\int_0^\infty q_t(a+e)\cdot V'_{t+1}(b+\eta e)de\end{aligned}$$

We also have

$$\frac{\partial Q_{t,2}}{\partial a}(b,a) = V_{t+1}(0)\cdot q_t(a-b)$$

and

$$\frac{\partial Q_{t,3}}{\partial a}(b,a) = b\cdot\int_0^1 q'_t(a-be)\cdot V_{t+1}(b(1-e))de$$

We integrate by parts, integrating $e \mapsto bq_t'(a-be)$ and differentiating $e \mapsto V_{t+1}(b(1-e))$.

$$\frac{\partial Q_{t,3}}{\partial a}(b,a) = [-q_t(a-be)\cdot V_{t+1}(b(1-e))]_0^1 - b\int_0^1 q_t(a-be)\cdot V_{t+1}'(b(1-e))de$$
$$\frac{\partial Q_{t,3}}{\partial a}(b,a) = q_t(a)V_{t+1}(b) - q_t(a-b)V_{t+1}(0) - b\int_0^1 q_t(a-be)\cdot V_{t+1}'(b(1-e))de$$

Finally

$$\begin{aligned}\frac{\partial Q_t}{\partial a}(b,a) &= c^+ - (c^+ + c^-)\cdot F_t(a-b)\\ &\quad - q_t(a)\cdot V_{t+1}(b) - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}'(b+\eta e)de\\ &\quad + V_{t+1}(0)\cdot q_t(a-b)\\ &\quad + q_t(a)V_{t+1}(b) - q_t(a-b)V_{t+1}(0) - b\int_0^1 q_t(a-be)\cdot V_{t+1}'(b(1-e))de\\ \frac{\partial Q_t}{\partial a}(b,a) &= c^+ - (c^+ + c^-)\cdot F_t(a-b) - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}'(b+\eta e)de\\ &\quad - b\int_0^1 q_t(a-be)\cdot V_{t+1}'(b(1-e))de\end{aligned}$$

We compute now the second order differentiates.

$$\frac{\partial^2 r_t}{\partial a^2}(b,a) = -(c^+ + c^-)q_t(a-b)$$

The differentiate along a of the first integral in $\frac{\partial Q_t}{\partial a}(b,a)$ can be obtained through the same techniques that we use to compute $\frac{\partial Q_{t,1}}{\partial a}(b,a)$ with V_{t+1}' instead of V_{t+1}. In addition, for similar reasons, all partial differentiates exist and are continuous, which makes Q_t of class $\mathcal{C}^2$.

$$\frac{\partial}{\partial a}\left(\int_0^\infty q_t(a+e)\cdot V_{t+1}'(b+\eta e)de\right) = -q_t(a)\cdot V_{t+1}'(b) - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}''(b+\eta e)de$$

The differentiate along a of the second integral is also similar to the differentiation along a of $Q_{t,3}$.

$$\begin{aligned}\frac{\partial}{\partial a}\left(b\int_0^1 q_t(a-be)\cdot V_{t+1}'(b(1-e))\right) &= q_t(a)V_{t+1}'(b) - q_t(a-b)V_{t+1}'(0)\\ &\quad - b\int_0^1 q_t(a-be)\cdot V_{t+1}''(b(1-e))de\end{aligned}$$

This finally leads to

$$\begin{aligned}\frac{\partial^2 Q_t}{\partial a^2}(b,a) &= -(c^+ + c^-)q_t(a-b)\\ &\quad - \eta\cdot\left(-q_t(a)\cdot V_{t+1}'(b) - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}''(b+\eta e)de\right)\\ &\quad - \left(q_t(a)V_{t+1}'(b) - q_t(a-b)V_{t+1}'(0) - b\int_0^1 q_t(a-be)\cdot V_{t+1}''(b(1-e))de\right)\\ \frac{\partial^2 Q_t}{\partial a^2}(b,a) &= \left(V_t'(0) - (c^+ + c^-)\right)\cdot q_t(a-b) - (1-\eta)\cdot q_t(a)\cdot V_{t+1}'(b)\\ &\quad + \eta^2\int_0^\infty q_t(a+e)\cdot V_{t+1}''(b+\eta e)de\\ &\quad + b\int_0^1 q_t(a-be)\cdot V_{t+1}''(b(1-e))de\end{aligned}$$

We compute now $\frac{\partial^2 Q_t}{\partial b^2}(b,a)$.

$$\frac{\partial^2 r_t}{\partial b^2}(b,a) = -(c^+ + c^-)q_t(a-b)$$

$$\frac{\partial}{\partial b}\left(b \cdot \int_0^1 q_t(a-be)\cdot V'_{t+1}(b(1-e))de\right) = b\int_0^1 q_t(a+be)\cdot V''_{t+1}(b(1-e))de + q_t(a-b)\cdot V'_{t+1}(0)$$

Therefore

$$\begin{aligned}
\frac{\partial^2 Q_t}{\partial b^2}(b,a) &= -(c^+ + c^-)q_t(a-b) \\
&\quad + \int_0^\infty q_t(a+e)\cdot V''_{t+1}(b+\eta e)de \\
&\quad + b\int_0^1 q_t(a-be)\cdot V''_{t+1}(b(1-e))de + q_t(a-b)\cdot V'_{t+1}(0) \\
\frac{\partial^2 Q_t}{\partial b^2}(b,a) &= \left(V'_{t+1}(0) - (c^+ + c^-)\right)\cdot q_t(a-b) \\
&\quad + \int_0^\infty q_t(a+e)\cdot V''_{t+1}(b+\eta e)de \\
&\quad + b\int_0^1 q_t(a-be)\cdot V''_{t+1}(b(1-e))de
\end{aligned}$$

Finally,

$$\begin{aligned}
\frac{\partial^2 Q_t}{\partial a \partial b}(b,a) &= \frac{\partial}{\partial a}\left(\frac{\partial Q_t}{\partial b}(b,a)\right) \\
&= (c^+ + c^-)\cdot q_t(a-b) \\
&\quad + \frac{\partial}{\partial a}\left(\int_0^\infty q_t(a+e)\cdot V'_{t+1}(b+\eta e)de\right) \\
&\quad + \frac{\partial}{\partial a}\left(b\int_0^1 q_t(a-be)\cdot V'_{t+1}(b(1-e))de\right) \\
&= (c^+ + c^-)\cdot q_t(a-b) \\
&\quad - q_t(a)\cdot V_{t+1}(b) - \eta\int_0^\infty q_t(a+e)\cdot V''_{t+1}(b+\eta e)de \\
&\quad + q_t(a)V_{t+1}(b) - q_t(a-b)V_{t+1}(0) - b\int_0^1 q_t(a-be)\cdot V''_{t+1}(b(1-e))de \\
\frac{\partial^2 Q_t}{\partial a \partial b}(b,a) &= \left((c^+ + c^-) - V'_t(0)\right)\cdot q_t(a-b) \\
&\quad - \eta\int_0^\infty q_t(a+e)\cdot V''_{t+1}(b+\eta e)de \\
&\quad - b\int_0^1 q_t(a-be)\cdot V''_{t+1}(b(1-e))de
\end{aligned}$$

Case $a \leq b$

$$\begin{aligned}\frac{\partial Q_{t,4}}{\partial a}(b,a) &= \int_0^\infty q_t'(a+e)\cdot V_{t+1}(b+\eta e)de \\ &= [q_t(a+e)\cdot V_{t+1}(b+\eta e)]_0^\infty - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}'(b+\eta e)de \\ \frac{\partial Q_{t,4}}{\partial a}(b,a) &= -q_t(a)\cdot V_{t+1}(b) - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}'(b+\eta e)de\end{aligned}$$

$$\begin{aligned}\frac{\partial Q_{t,5}}{\partial a}(b,a) &= \int_0^1 q_t(ae)\cdot V_{t+1}(b-a(1-e))de \\ &\quad + a\int_0^1 \left(eq_t'(ae)\cdot V_{t+1}(b-a(1-e))de + q_t(ae)\cdot(e-1)V_{t+1}'(b-a(1-e))\right)de \\ &= \int_0^1 q_t(ae)\cdot V_{t+1}(b-a(1-e))de \\ &\quad + [eq_t(ae)V_{t+1}(b-a(1-e))]_0^1 - \int_0^1 q_t(ae)\cdot V_{t+1}(b-a(1-e))de \\ &\quad - a\int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e))de \\ \frac{\partial Q_{t,5}}{\partial a}(b,a) &= q_t(a)V_{t+1}(b) - a\int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e))de\end{aligned}$$

All put together

$$\begin{aligned}\frac{\partial Q_t}{\partial a}(b,a) &= c^+ \\ &\quad - q_t(a)\cdot V_{t+1}(b) - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}'(b+\eta e)de \\ &\quad + q_t(a)V_{t+1}(b) - a\int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e))de \\ \frac{\partial Q_t}{\partial a}(b,a) &= c^+ - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}'(b+\eta e)de - a\int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e))de\end{aligned}$$

$$\frac{\partial^2 Q_t}{\partial a^2}(b,a) = \eta^2\int_0^\infty q_t(a+e)\cdot V_{t+1}''(b+\eta e)de + a\int_0^1 q_t(ae)\cdot V_{t+1}''(b-a(1-e))de$$

$$\frac{\partial Q_t}{\partial b}(b,a) = \int_0^\infty q_t(a+e)\cdot V_{t+1}'(b+\eta e)de + a\int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e))de$$

$$\frac{\partial^2 Q_t}{\partial b^2}(b,a) = \int_0^\infty q_t(a+e)\cdot V_{t+1}''(b+\eta e)de + a\int_0^1 q_t(ae)\cdot V_{t+1}''(b-a(1-e))de$$

$$
\begin{aligned}
\frac{\partial^2 Q_t}{\partial a \partial b}(b,a) &= \int_0^\infty q_t'(a+e)\cdot V_{t+1}'(b+\eta e)de \\
&\quad + \int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e))de \\
&\quad + a\int_0^1 (eq_t'(ae)\cdot V_{t+1}'(b-a(1-e)) - (1-e)q_t(ae)\cdot V_{t+1}''(b-a(1-e)))de \\
&= \left[q_t(a+e)\cdot V_{t+1}'(b+\eta e)\right]_0^\infty - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}''(b+\eta e)de \\
&\quad + \int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e))de \\
&\quad + \int_0^1 (eaq_t'(ae)\cdot V_{t+1}'(b-a(1-e)) + eq_t(ae)\cdot aV_{t+1}''(b-a(1-e)))de \\
&\quad - a\int_0^1 q_t(ae)\cdot V_{t+1}''(b-a(1-e))de \\
&= -q_t(a)\cdot V_{t+1}'(b) - \eta\int_0^\infty q_t(a+e)\cdot V_{t+1}''(b+\eta e)de \\
&\quad + \int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e))de \\
&\quad + \left[eq_t(ae)\cdot V_{t+1}'(b-a(1-e))\right]_0^1 - \int_0^1 q_t(ae)\cdot V_{t+1}'(b-a(1-e)) \\
&\quad - a\int_0^1 q_t(ae)\cdot V_{t+1}''(b-a(1-e))de \\
\frac{\partial^2 Q_t}{\partial a \partial b}(b,a) &= -\eta\int_0^\infty q_t(a+e)\cdot V_{t+1}''(b+\eta e)de - a\int_0^1 q_t(ae)\cdot V_{t+1}''(b-a(1-e))de
\end{aligned}
$$

A.1.2 Extension of the proof of lemma 12

Let $(b, a, h) \in \mathcal{B} \times \mathcal{A} \times \mathbb{R}^+$. We have that

$$
\begin{aligned}
&\frac{\partial Q_t}{\partial a}(b+\eta h, a+h) - \frac{\partial Q_t}{\partial a}(b,a) \\
=&c^+ - (c^+ + c^-) \cdot F_t(a-b+(1-\eta)h) \\
&- \eta \int_{a+h}^{\infty} q_t(e) \cdot V'_{t+1}(b+\eta(e-a))de \\
&- \int_{a-b+(1-\eta)h}^{a+h} q_t(e) \cdot V'_{t+1}(b+e-a-(1-\eta)h)de \\
&- c^+ + (c^+ + c^-) \cdot F_t(a-b) \\
&+ \eta \int_{a}^{\infty} q_t(e) \cdot V'_{t+1}(b+\eta(e-a))de \\
&+ \int_{a-b}^{a} q_t(e) \cdot V'_{t+1}(b+e-a)de \\
=&- \int_{a-b}^{a-b+(1-\eta)h} q_t(e)(c^+ + c^-)de \\
&+ \eta \int_{a}^{a+h} q_t(e) \cdot V'_{t+1}(b+\eta(e-a))de \\
&- \int_{a}^{a+h} q_t(e) \cdot V'_{t+1}(b+e-a-(1-\eta)h)de \\
&- \int_{a-b+(1-\eta)h}^{a} q_t(e) \cdot \left(V'_{t+1}(b+e-a-(1-\eta)h) - V'_{t+1}(b+e-a)\right) de \\
&+ \int_{a-b}^{a-b+(1-\eta)h} q_t(e) \cdot V'_{t+1}(b+e-a)de \\
=&- \int_{a-b}^{a-b+(1-\eta)h} q_t(e) \left((c^+ + c^-) - V'_{t+1}(b+e-a)\right) de \\
&- \int_{a}^{a+h} q_t(e) \cdot \left(V'_{t+1}(b+e-a-(1-\eta)h) - \eta V'_{t+1}(b+\eta(e-a))\right) de \\
&- \int_{a-b+(1-\eta)h}^{a} q_t(e) \cdot \left(V'_{t+1}(b+e-a-(1-\eta)h) - V'_{t+1}(b+e-a)\right) de
\end{aligned}
$$

A.1.3 Corollary of lemma 12

Corollary 1. *If $(\mathcal{H}_{t+1})$ is true and if there exists $(b_0, a_0, a'_0) \in \mathcal{B} \times \mathcal{A}^2$ such that $a_0 < a'_0$ and*

$$\forall a \in [a_0, a'_0], \frac{\partial Q_t}{\partial a}(b,a) = 0$$

then

$$\forall b \in [b_0, b_0 + \eta \cdot (a'_0 - a_0)], \forall a \in [a_0 + \eta^{-1} \cdot (b - b_0), a'_0], \quad \frac{\partial Q_t}{\partial a}(b,a) = 0$$

and

$$\forall b \in [b_0 - \eta \cdot (a'_0 - a_0), b_0], \forall a \in [a_0, a'_0 - \eta^{-1} \cdot (b_0 - b)], \quad \frac{\partial Q_t}{\partial a}(b,a) = 0$$

Proof. We assume that $(\mathcal{H}_{t+1})$ is true and that there exists $(b_0, a_0, a_0') \in \mathcal{B} \times \mathcal{A}^2$ such that $a_0 < a_0'$

$$\forall a \in [a_0, a_0'], \frac{\partial Q_t}{\partial a}(b, a) = 0,$$

and we will show that

$$\forall h \in \mathbb{R}^+, \quad h \leq a_0' - a_0 \Rightarrow \frac{\partial Q_t}{\partial a}(b_0 + \eta \cdot h, a_0 + h) = 0.$$

Let $h \in]0, a_0' - a_0]$. According to eq. (6.21), we have $\frac{\partial Q_t}{\partial a}(b + \eta h, a + h) \leq \frac{\partial Q_t}{\partial a}(b, a)$. By hypothesis, $\frac{\partial Q_t}{\partial a}(b, a + h) = \frac{\partial Q_t}{\partial a}(b, a)$, thus $\frac{\partial Q_t}{\partial a}(b + \eta h, a + h) \leq \frac{\partial Q_t}{\partial a}(b, a + h)$. Since we know that $\frac{\partial^2 Q_t}{\partial b \partial a}(b, a) \geq 0$, we get $\frac{\partial Q_t}{\partial a}(b + \eta h, a + h) \geq \frac{\partial Q_t}{\partial a}(b, a + h)$. As a consequence, $\frac{\partial Q_t}{\partial a}(b + \eta h, a + h) = \frac{\partial Q_t}{\partial a}(b, a + h) = 0$.

This holds for any $h \in]0, a_0' - a_0]$, hence the first part of the corollary. The same line of reasoning with $(b - h, a - h)$ instead of $(b + h, a + h)$ and $(b - \eta h, a - h)$ instead of $(b - \eta h, a + h)$, leads to the second part of the corollary. □

A.1.4 Extension of the proof of lemma 14

We assume that $(\mathcal{H}_{t+1})$ is true. Let $b \in \mathcal{B}$ such that $\frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) = 0$. We show that V_t' is differentiable in b by demonstrating that for any $\varepsilon \in \mathbb{R}_\star^+$, there exists $r \in \mathbb{R}_\star^+$ such that for all $b' \in \mathcal{B}$,

$$|b - b'| < r \Rightarrow \left| \frac{\frac{Q_t(b, m(b)) - Q_t(b', m(b'))}{b - b'} - \frac{\partial Q_t}{\partial b}(b, m(b))}{b - b'} - \frac{\partial^2 Q_t}{\partial b^2}(b, m(b)) \right| < \varepsilon.$$

Let $\varepsilon \in \mathbb{R}_\star^+$. Q_t is twice differentiable, hence for all $a \in \mathcal{A}$ there exists r_1 such that

$$\forall b' \in \mathcal{B}, |b - b'| < r_2 \Rightarrow \left| \frac{\frac{Q_t(b, a) - Q_t(b', a)}{b - b'} - \frac{\partial Q_t}{\partial b}(b, a)}{b - b'} - \frac{\partial^2 Q_t}{\partial b^2}(b, a) \right| < \frac{\varepsilon}{2}.$$

In addition, for all $(b', a') \in \mathcal{B} \times \mathcal{A}$ there exists r_2 such that

$$\forall a \in \mathcal{A}, |a - a'| < r_2 \Rightarrow \left| \frac{\frac{Q_t(b', a) - Q_t(b', a')}{a - a'} - \frac{\partial Q_t}{\partial a}(b', a)}{a - a'} - \frac{\partial^2 Q_t}{\partial a^2}(b', a) \right| < \frac{\eta^2 \varepsilon}{2}.$$

Especially, for all $(b', a') \in \mathcal{B} \times \mathcal{A}$

$$\begin{aligned} |m(b') - a'| < r_2 &\Rightarrow \left| \frac{\frac{Q_t(b', m(b')) - Q_t(b', a')}{m(b') - a'} - \frac{\partial Q_t}{\partial a}(b', m(b'))}{m(b') - a'} - \frac{\partial^2 Q_t}{\partial a^2}(b', m(b')) \right| < \frac{\eta^2 \varepsilon}{2} \\ &\Rightarrow \left| \frac{Q_t(b', m(b')) - Q_t(b', a')}{(m(b') - a')^2} \right| < \frac{\eta^2 \varepsilon}{2}, \end{aligned}$$

which means for all $b' \in \mathcal{B}$

$$|m(b') - m(b)| < r_2 \Rightarrow \left| \frac{Q_t(b', m(b')) - Q_t(b', m(b))}{(m(b') - m(b))^2} \right| < \frac{\eta^2 \varepsilon}{2}.$$

We have seen in the proof of lemma 13 that, according to lemma 12,

$$\forall (b_1, b_2) \in \mathcal{B}^2, |b_1 - b_2| < \eta \cdot r_2 \Rightarrow |m(b_1) - m(b_2)| < r_2.$$

This leads to

$$\begin{aligned}\forall b' \in \mathcal{B}, |b-b'| < \eta \cdot r_2 \Rightarrow & \left|m(b) - m(b')\right| < r_2 \\ \Rightarrow & \left|\frac{Q_t(b', m(b')) - Q_t(b', m(b))}{(m(b') - m(b))^2}\right| < \frac{\eta^2 \varepsilon}{2}.\end{aligned}$$

Altogether, for all $b' \in \mathcal{B}$, if $|b-b'| < \eta \cdot \min(r_1, r_2)$ then

$$\begin{aligned}\left|\frac{\frac{Q_t(b,m(b)) - Q_t(b',m(b'))}{b-b'} - \frac{\partial Q_t}{\partial b}(b, m(b))}{b-b'} - \frac{\partial^2 Q_t}{\partial b^2}(b, m(b))\right| & < \frac{\varepsilon}{2} + \frac{\eta^2 \varepsilon}{2} \times \frac{1}{\eta^2} \\ & < \varepsilon.\end{aligned}$$

As this holds for any $\varepsilon \in \mathbb{R}_\star^+$, V_t' is differentiable in b.

A.1.5 Extension of the proof of lemma 15

We know that there exists a neighborhood containing $(b_0, m(b_0))$ where m is of class $\mathcal{C}^1$. Within this neighborhood, the function $b \mapsto \frac{\partial Q_t}{\partial a}(b, m(b))$ is null, its differentiate is null as well, therefore

$$\frac{\partial^2 Q_t}{\partial b \partial a}(b, m(b)) + m'(b) \cdot \frac{\partial^2 Q_t}{\partial a^2}(b, m(b)) = 0.$$

We can then add freely the left member to the expression of $V_t''(b_0)$.

$$\begin{aligned}V_t''(b_0) &= \frac{\partial^2 Q_t}{\partial b^2}(b_0, m(b_0)) + m'(b_0) \cdot \frac{\partial^2 Q_t}{\partial a^2}(b_0, m(b_0)) + (1 + m'(b_0)) \cdot \frac{\partial^2 Q_t}{\partial a \partial b}(b_0, m(b_0)) \\ &= \Big(m'(0) - (c^+ + c^-)\Big) \cdot q_t(a - b_0) + \int_0^\infty q_t(a+e) \cdot V_{t+1}''(b_0 + \eta e) de \\ &\quad + b \int_0^1 q_t(a - b_0 e) \cdot V_{t+1}''(b_0(1-e)) de \\ &\quad + m'(b_0) \cdot \Bigg(\Big(m'(0) - (c^+ + c^-)\Big) \cdot q_t(a - b_0) - (1-\eta) \cdot q_t(a) \cdot V_{t+1}'(b_0) \\ &\quad + \eta^2 \int_0^\infty q_t(a+e) \cdot V_{t+1}''(b_0 + \eta e) de + b \int_0^1 q_t(a - b_0 e) \cdot V_{t+1}''(b_0(1-e)) de \Bigg) \\ &\quad + (1 + m'(b_0)) \cdot \Bigg(\Big((c^+ + c^-) - m'(0)\Big) \cdot q_t(a - b_0) \\ &\quad - \eta \int_0^\infty q_t(a+e) \cdot V_{t+1}''(b_0 + \eta e) de - b \int_0^1 q_t(a - b_0 e) \cdot V_{t+1}''(b_0(1-e)) de \Bigg) \\ &= -(1-\eta) \cdot q_t(a) \cdot V_{t+1}'(b_0) \cdot m'(b_0) \\ &\quad + (\eta^2 + V_{t+1}'(b_0) - (1 + m'(b_0)) \cdot \eta) \cdot \int_0^\infty q_t(a+e) \cdot V_{t+1}''(b_0 + \eta e) de \\ &= -(1-\eta) \cdot q_t(a) \cdot V_{t+1}'(b_0) \cdot m'(b_0) \\ &\quad + (1 + \eta^2 \cdot m'(b_0) - (1 + m'(b_0)) \cdot \eta) \cdot \int_0^\infty q_t(a+e) \cdot V_{t+1}''(b_0 + \eta e) de \\ V_t''(b_0) &= -(1-\eta) \cdot q_t(a) \cdot V_{t+1}'(b_0) \cdot m'(b_0) \\ &\quad + (1-\eta)(1 - \eta \cdot m'(b_0)) \cdot \int_0^\infty q_t(a+e) \cdot V_{t+1}''(b_0 + \eta e) de\end{aligned}$$

A.1.6 Extension of the proof of theorem 3

If $\pi_t(b) \leq b$, then according to eq. (6.18),

$$a \int_0^1 q_t(ae) \cdot V'_{t+1}(b - a(1-e))de = c^+ - \eta \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b + \eta e)de$$

and according to eqs. (6.20) and (6.22),

$$V'_t(b) = \int_0^\infty q_t(a+e) \cdot V'_{t+1}(b+\eta e)de + a \int_0^1 q_t(ae) \cdot V'_{t+1}(b - a(1-e))de,$$

hence

$$V'_t(b) = c^+ + (1-\eta) \cdot \int_0^\infty q_t(\pi_t(b) + e) \cdot V'_{t+1}(b + \eta e)de.$$

A.2 Bounded batteries and unbounded workload

We develop here the Imperfect-BatteryBounded model, in which the battery has a maximum capacity, *i.e.* $\mathcal{B} = \{0, \dots, B^{\max}\}$. The model includes two gears, whether the battery contains energy to cover the accepted work or not.

Case $a \geq b$ We decompose $Q_t(b, a)$ almost in the same way as in the Imperfect-Unbounded model, according to the renewable energy e:

- the reward remains unchanged,
- if $b + e < a$, there is not enough energy, and the battery will be empty at the next decision,
- if $a - b < e < a$, the renewable energy is not sufficient, and the battery is discharged to run the accepted work,
- if $a < e < a + (B^{\max} - b)/\eta$, the accepted work can be fully run with renewable energy, and the whole excess energy is stored in the battery,
- if $b + \eta(e - a) > B^{\max}$, the battery cannot store all the energy, and the battery is subsequently full.

We employ the same techniques as before to obtain

$$\begin{aligned} Q_t(b,a) = a \cdot c^+ - (c^+ + c^-) \cdot (a-b) \cdot \int_0^1 F_t((a-b)e)de + V_{t+1}(0) \cdot F_t(a-b) \\ + b \cdot \int_0^1 q_t(a - be) \cdot V_{t+1}(b(1-e))de \\ + \eta^{-1}(B^{\max} - b) \int_0^1 q_t\left(\eta^{-1}(B^{\max} - b)e + a\right) V_{t+1}(b - (B^{\max} - b)e)de \\ + (1 - F_t(a + \eta^{-1}(B^{\max} - b)))V_{t+1}(B^{\max}). \end{aligned}$$

The same arguments that we used when the battery was not bounded still hold and prove that Q_t is of class $\mathcal{C}^2$. In addition, we can differentiate under the integral, and obtain the following similar partial differentiates.

$$\begin{aligned} \frac{\partial Q_t}{\partial b}(b,a) = (c^+ + c^-) \cdot F_t(a-b) + b \int_0^1 q_t(a - be) \cdot V'_{t+1}(b(1-e))de \\ + \eta^{-1}(B^{\max} - b) \int_0^1 q_t\left(\eta^{-1}(B^{\max} - b)e + a\right) V'_{t+1}(b - (B^{\max} - b)e)de \end{aligned}$$

$$\frac{\partial Q_t}{\partial a}(b,a) = c^+ - (c^+ + c^-) \cdot F_t(a-b) - b\int_0^1 q_t(a-be) \cdot V'_{t+1}(b(1-e))de \\ - (B^{\max} - b)\int_0^1 q_t\left(\eta^{-1}(B^{\max}-b)e + a\right) V'_{t+1}(b - (B^{\max}-b)e)de$$

When it comes to the demonstration of the bounds, either on the differentiate of V_t (lemmas 16 and 17) or on our optimal policy (lemma 12), the following expression shall be preferred

$$\frac{\partial Q_t}{\partial a}(b,a) = c^+ - (c^+ + c^-) \cdot F_t(a-b) - \int_{a-b}^{a} q_t(e) \cdot V'_{t+1}(b+e-a)de \\ - \eta\int_a^{a+\eta^{-1}(B^{\max}-b)} q_t(e)V'_{t+1}(b+(e-a)\eta)de$$

$$\frac{\partial^2 Q_t}{\partial a^2}(b,a) = \left(V'_t(0) - (c^+ + c^-)\right) \cdot q_t(a-b) - (1-\eta) \cdot q_t(a) \cdot V'_{t+1}(b) \\ - \eta \cdot q_t(\eta^{-1}(B^{\max}-b)+a) \cdot V'_{t+1}(B^{\max}) \\ + \eta\int_0^1 q_t(\eta^{-1}(B^{\max}-b)e + a)V''_{t+1}(b-(B^{\max}-b)e)de \\ + b\int_0^1 q_t(a-be) \cdot V''_{t+1}(b(1-e))de$$

$$\frac{\partial^2 Q_t}{\partial b^2}(b,a) = \left(V'_{t+1}(0) - (c^+ + c^-)\right) \cdot q_t(a-b) \\ - \eta^{-1} q_t(\eta^{-1}(B^{\max}-b)+a) \cdot V'_{t+1}(B^{\max}) \\ + \eta^{-1}(B^{\max}-b)\int_0^1 q_t(\eta^{-1}(B^{\max}-b)e + a)V''_{t+1}(b-(B^{\max}-b)e)de \\ + b\int_0^1 q_t(a-be) \cdot V''_{t+1}(b(1-e))de$$

$$\frac{\partial^2 Q_t}{\partial a \partial b}(b,a) = \left((c^+ + c^-) - V'_t(0)\right) \cdot q_t(a-b) \\ + q_t(\eta^{-1}(B^{\max}-b)+a) \cdot V'_{t+1}(B^{\max}) \\ - (B^{\max}-b)\int_0^1 q_t(\eta^{-1}(B^{\max}-b)e + a)V''_{t+1}(b-(B^{\max}-b)e)de \\ - b\int_0^1 q_t(a-be) \cdot V''_{t+1}(b(1-e))de$$

Case $a \leq b$ The accepted work is certainly executed. The battery is discharged only when $e < a$, and part of the renewable energy is lost when $e > a + \eta^{-1}(B^{\max} - b)$.

$$Q_t(b,a) = a \cdot c^+ + a\int_0^1 q_t(ae) \cdot V_{t+1}(b - a(1-e))de \\ + \eta^{-1}(B^{\max}-b)\int_0^1 q_t\left(\eta^{-1}(B^{\max}-b)e + a\right) V_{t+1}(b-(B^{\max}-b)e)de \\ + (1 - F_t(a + \eta^{-1}(B^{\max}-b)))V_{t+1}(B^{\max})$$

$$\frac{\partial Q_t}{\partial a}(b,a) = c^+ - (B^{\max}-b)\int_0^1 q_t\left(\eta^{-1}(B^{\max}-b)e+a\right)V'_{t+1}(b-(B^{\max}-b)e)de \\ - a\int_0^1 q_t(ae)\cdot V'_{t+1}(b-a(1-e))de$$

$$\frac{\partial Q_t}{\partial b}(b,a) = \eta^{-1}(B^{\max}-b)\int_0^1 q_t\left(\eta^{-1}(B^{\max}-b)e+a\right)V'_{t+1}(b-(B^{\max}-b)e)de \\ + a\int_0^1 q_t(ae)\cdot V'_{t+1}(b-a(1-e))de$$

$$\frac{\partial^2 Q_t}{\partial a^2}(b,a) = q_t(\eta^{-1}(B^{\max}-b)+a)\cdot V'_{t+1}(B^{\max}) \\ - (B^{\max}-b)\int_0^1 q_t(\eta^{-1}(B^{\max}-b)e+a)V''_{t+1}(b-(B^{\max}-b)e)de \\ + a\int_0^1 q_t(ae)\cdot V''_{t+1}(b-a(1-e))de$$

$$\frac{\partial^2 Q_t}{\partial b^2}(b,a) = \int_0^\infty q_t(a+e)\cdot V''_{t+1}(b+\eta e)de + a\int_0^1 q_t(ae)\cdot V''_{t+1}(b-a(1-e))de$$

A.3 Approximations

For a given $(b,a)\in\mathcal{B}\times\mathcal{A}$, we find an approximation of

$$P_t(b,a) = V_{t+1}^{app}(0)\cdot F_t(a-b) + (1-F_t(a+\eta^{-1}(B^{\max}-b)))V_{t+1}^{app}(B^{\max}) \\ + \int_{a-b}^{a} q_t(e)\cdot V_{t+1}^{app}(b+e-a)de + \int_a^{a+\eta^{-1}(B^{\max}-b)} q_t(e)\cdot V_{t+1}^{app}(b+\eta(e-a))de.$$

We define ϕ a bijective function from $[a-b, a+\eta^{-1}(B^{\max}-b)]$ to $[0,B^{\max}]$ as

$$\phi(e) = \begin{cases} b+e-a & \text{if } e<a \\ b+\eta(e-a) & \text{otherwise} \end{cases}$$

so that

$$P_t(b,a) = V_{t+1}^{app}(0)\cdot F_t(a-b) + (1-F_t(a+\eta^{-1}(B^{\max}-b)))V_{t+1}^{app}(B^{\max}) \\ + \int_{a-b}^{a+\eta^{-1}(B^{\max}-b)} q_t(e)\cdot V_{t+1}^{app}(\phi(e))de$$

We assume that we have a piecewise constant discretization of V_{t+1}^{app}: there exist $\varepsilon^{(2)}$, $(u_i)_{i\in\{0,\dots,N_V\}}$ and $(\widetilde{V}_i)_{i\in\{0,\dots,N_V\}}$ such that for all $i\in\{0,\dots,N_V\}$, $\widetilde{V}_i = V_{t+1}^{app}(u_i)$ and if $i<N_V$, then $0<\widetilde{V}_{i+1}-\widetilde{V}_i<\varepsilon^{(2)}$.

Let ℓ_t^{alg} defined as

$$\ell_t^{alg}(b,a) = \widetilde{V}_0\cdot F_t(a-b) + (1-F_t(a+\eta^{-1}(B^{\max}-b)))\widetilde{V}_{N_v} \\ + \sum_{i=0}^{N_V-1}(F_t(\phi^{-1}(u_{i+1})) - F_t(\phi^{-1}(u_i)))\widetilde{V}_i.$$

We show that ℓ_t^{alg} is not further than $\varepsilon^{(2)}$ from V_{t+1}^{app}. We can first let appear the discretization steps in P_t

$$P_t(b,a) = V_{t+1}^{app}(0) \cdot F_t(a-b) + (1 - F_t(a + \eta^{-1}(B^{\max} - b)))V_{t+1}^{app}(B^{\max}) \\ + \sum_{i=0}^{N_V-1} \int_{\phi(u_i)}^{\phi(u_{i+1})} q_t(e) \cdot V_{t+1}^{app}(\phi(e))de,$$

which leads by changing the variables with $e \leftarrow \phi(e)$

$$P_t(b,a) = V_{t+1}^{app}(0) \cdot F_t(a-b) + (1 - F_t(a + \eta^{-1}(B^{\max} - b)))V_{t+1}^{app}(B^{\max}) \\ + \sum_{i=0}^{N_V-1} \int_{u_i}^{u_{i+1}} \frac{1}{\phi'(\phi^{-1}(e))} \cdot q_t(\phi^{-1}(e)) \cdot V_{t+1}^{app}(e)de.$$

The distance between the approximation and P_t is

$$\begin{aligned} |P_t(b,a) - \ell_t^{alg}(b,a)| &= \sum_{i=0}^{N_V-1} \int_{u_i}^{u_{i+1}} \phi'(e) \cdot q_t(\phi^{-1}(e)) \cdot |V_{t+1}^{app}(e) - V_{t+1}^{app}(u_i)|de \\ &< \varepsilon^{(2)}(F_t(a + \eta^{-1}(B^{\max} - b)) - F_t(a-b)) \\ |P_t(b,a) - \ell_t^{alg}(b,a)| &< \varepsilon^{(2)} \end{aligned}$$

Bibliography

[1] Parallel Workloads Archive: Standard Workload Format. http://www.cs.huji.ac.il/labs/parallel/workload/swf.html, 2006.

[2] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering – A decade review. *Information Systems*, 53:16–38, October 2015. ISSN 0306-4379. doi: 10.1016/j.is.2015.04.007.

[3] Taimoor Akhtar and Christine A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. *Journal of Global Optimization*, 64(1):17–32, January 2016. ISSN 0925-5001, 1573-2916. doi: 10.1007/s10898-015-0270-y.

[4] Baris Aksanli, Jagannathan Venkatesh, Liuyi Zhang, and Tajana Rosing. Utilizing Green Energy Prediction to Schedule Mixed Batch and Service Jobs in Data Centers. *SIGOPS Oper. Syst. Rev.*, 45(3):53–57, January 2012. ISSN 0163-5980. doi: 10.1145/2094091.2094105.

[5] Abdel-Rahman Al-Ghuwairi and Jonathan Cook. Modeling and enforcement of cloud computing service level agreements. Technical report, 2012.

[6] M. Alam, K. A. Shakil, and S. Sethi. Analysis and Clustering of Workload in Google Cluster Trace Based on Resource Usage. In *Proceedings of the 2016 IEEE Intl Conference on Computational Science and Engineering (CSE)*, pages 740–747, August 2016. doi: 10.1109/CSE-EUC-DCABES.2016.271.

[7] Erik Alsema. Energy Payback Time and CO2 Emissions of PV Systems. In Augustin McEvoy, Tom Markvart, and Luis Castañer, editors, *Practical Handbook of Photovoltaics*, pages 1097–1117. Academic Press, Boston, second edition edition, January 2012. ISBN 978-0-12-385934-1. doi: 10.1016/B978-0-12-385934-1.00037-4.

[8] Anders S. G. Andrae and Tomas Edler. On Global Electricity Usage of Communication Technology: Trends to 2030. *Challenges*, 6(1):117–157, April 2015. doi: 10.3390/challe6010117.

[9] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de-Pison, and F. Antonanzas-Torres. Review of photovoltaic power forecasting. *Solar Energy*, 136:78–111, October 2016. ISSN 0038-092X. doi: 10.1016/j.solener.2016.06.069.

[10] Martin Arlitt and Tai Jin. A workload characterization study of the 1998 world cup web site. *IEEE network*, 14(3):30–37, 2000.

[11] Y. M. Atwa, E. F. El-Saadany, M. M. A. Salama, and R. Seethapathy. Optimal Renewable Resources Mix for Distribution System Energy Loss Minimization. *IEEE*

Transactions on Power Systems, 25(1):360–370, February 2010. ISSN 0885-8950. doi: 10.1109/TPWRS.2009.2030276.

[12] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, and Albert Zomaya. A taxonomy and survey of energy-efficient data centers and cloud computing systems. In *Advances in Computers*, volume 82, pages 47–111. Elsevier, 2011.

[13] Anne Benoit, Laurent Lefèvre, Anne-Cécile Orgerie, and Issam Raïs. Shutdown Policies with Power Capping for Large Scale Computing Systems. In Francisco F. Rivera, Tomás F. Pena, and José C. Cabaleiro, editors, *Euro-Par 2017: Parallel Processing*, Lecture Notes in Computer Science, pages 134–146. Springer International Publishing, 2017. ISBN 978-3-319-64203-1.

[14] Theophilus Benson, Aditya Akella, and David A. Maltz. Network Traffic Characteristics of Data Centers in the Wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 267–280, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0483-2. doi: 10.1145/1879141.1879175.

[15] Philip E. Bett and Hazel E. Thornton. The climatological relationships between wind and solar energy supply in Britain. *Renewable Energy*, 87:96–110, March 2016. ISSN 0960-1481. doi: 10.1016/j.renene.2015.10.006.

[16] Kashif Bilal, Saif Ur Rehman Malik, Osman Khalid, Abdul Hameed, Enrique Alvarez, Vidura Wijaysekara, Rizwana Irfan, Sarjan Shrestha, Debjyoti Dwivedy, Mazhar Ali, Usman Shahid Khan, Assad Abbas, Nauman Jalil, and Samee U. Khan. A taxonomy and survey on Green Data Center Networks. *Future Generation Computer Systems*, 36:189–208, July 2014. ISSN 0167-739X. doi: 10.1016/j.future.2013.07.006.

[17] M. G. Lucchesi Bosilovich. MERRA-2: File Specification. Technical report, September 2015.

[18] George EP Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[19] Gerd Bramerdorfer and Alexandru-Ciprian Zăvoianu. Surrogate-based multi-objective optimization of electrical machine designs facilitating tolerance analysis. *IEEE Transactions on Magnetics*, 53(8):1–11, 2017.

[20] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. Finding Knees in Multi-objective Optimization. In Xin Yao, Edmund K. Burke, José A. Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiňo, Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, Lecture Notes in Computer Science, pages 722–731. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-30217-9.

[21] Michael Brown and Jose Renau. Rerack: Power simulation for data centers with renewable energy generation. *ACM SIGMETRICS Performance Evaluation Review*, 39(3):77–81, 2011.

[22] Alexander E.I. Brownlee, John R. Woodward, and Jerry Swan. Metaheuristic Design Pattern: Surrogate Fitness Functions. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO

Companion '15, pages 1261–1264, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3488-4. doi: 10.1145/2739482.2768499.

[23] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 41(1):23–50, January 2011. ISSN 1097-024X. doi: 10.1002/spe.995.

[24] California Environmental Protection Agency. California GHG Emission Inventory Edition 2016, 2016.

[25] California Independant System Operator. Renewables Reporting. http://www.caiso.com/market/Pages/ReportsBulletins/DailyRenewablesWatch.aspx.

[26] H. Casanova. Simgrid: A toolkit for the simulation of application scheduling. In *First IEEE/ACM International Symposium on Cluster Computing and the Grid, 2001. Proceedings*, pages 430–437, 2001. doi: 10.1109/CCGRID.2001.923223.

[27] Stephane Caux, Paul Renaud-Goud, Gustavo Rostirolla, and Patricia Stolf. IT Optimization for Datacenters Under Renewable Power Constraint. In Marco Aldinucci, Luca Padovani, and Massimo Torquati, editors, *Euro-Par 2018: Parallel Processing*, Lecture Notes in Computer Science, pages 339–351. Springer International Publishing, 2018. ISBN 978-3-319-96983-1.

[28] Massimo Ceraolo. New dynamical models of lead-acid batteries. *IEEE transactions on Power Systems*, 15(4):1184–1190, 2000.

[29] F. K. P. Chan, A. W. C. Fu, and C. Yu. Haar wavelets for efficient similarity search of time-series: With and without time warping. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):686–705, May 2003. ISSN 1041-4347. doi: 10.1109/TKDE.2003.1198399.

[30] Kin-Pong Chan and Ada Wai-Chee Fu. Efficient time series matching by wavelets. In *Data Engineering, 1999. Proceedings., 15th International Conference On*, pages 126–133. IEEE, 1999.

[31] Muhammad Tayyab Chaudhry, Teck Chaw Ling, Atif Manzoor, Syed Asad Hussain, and Jongwon Kim. Thermal-Aware Scheduling in Green Data Centers. *ACM Comput. Surv.*, 47(3):39:1–39:48, February 2015. ISSN 0360-0300. doi: 10.1145/2678278.

[32] Changsong Chen, Shanxu Duan, Tao Cai, and Bangyin Liu. Online 24-h solar power forecasting based on weather type classification using artificial neural network. *Solar Energy*, 85(11):2856–2870, November 2011. ISSN 0038-092X. doi: 10.1016/j.solener.2011.08.027.

[33] K. H. Chen and Z. D. Ding. Lithium-ion battery lifespan estimation for hybrid electric vehicle. In *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pages 5602–5605, May 2015. doi: 10.1109/CCDC.2015.7161797.

[34] C.-F. Chiasserini and Ramesh R. Rao. Energy efficient battery management. *IEEE journal on selected areas in communications*, 19(7):1235–1245, 2001.

[35] Carla-Fabiana Chiasserini and Ramesh R. Rao. Pulsed battery discharge in communication devices. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 88–95. ACM, 1999.

[36] Daniel D. Chiras. *Power from the Sun: A Practical Guide to Solar Electricity*. New Society Publishers, October 2013. ISBN 978-1-55092-433-6.

[37] Yinghao Chu and Peter Meisen. Review and comparison of different solar energy technologies. *Global Energy Network Institute (GENI), San Diego, CA*, 2011.

[38] Leandro Cupertino, Georges Da Costa, Ariel Oleksiak, Wojciech Piatek, Jean-Marc Pierson, Jaume Salom, Laura Sisó, Patricia Stolf, Hongyang Sun, and Thomas Zilio. Energy-efficient, thermal-aware modeling and simulation of data centers: The CoolEmAll approach and evaluation results. *Ad Hoc Networks*, 25, Part B:535–553, February 2015. ISSN 1570-8705. doi: 10.1016/j.adhoc.2014.11.002.

[39] Leandro Fontoura Cupertino, Georges Da Costa, and Jean-Marc Pierson. Towards a generic power estimator. *Computer Science - Research and Development*, 30(2): 145–153, May 2015. ISSN 1865-2042. doi: 10.1007/s00450-014-0264-x.

[40] Claire Curry. Lithium-ion battery costs and market. *Bloomberg New Energy Finance*, 5, 2017.

[41] Georges Da Costa, Leo Grange, and Ines De Courchelle. Modeling and generating large-scale Google-like workload. In *2016 Seventh International Green and Sustainable Computing Conference (IGSC)*, pages 1–7, Hangzhou, China, 2016. IEEE. ISBN 978-1-5090-5117-5. doi: 10.1109/IGCC.2016.7892623.

[42] Georges Da Costa, Léo Grange, and Inès de Courchelle. Modeling, classifying and generating large-scale Google-like workload. *Sustainable Computing: Informatics and Systems*, 19:305–314, September 2018. ISSN 2210-5379. doi: 10.1016/j.suscom.2017.12.004.

[43] M. Dayarathna, Y. Wen, and R. Fan. Data Center Energy Consumption Modeling: A Survey. *IEEE Communications Surveys Tutorials*, 18(1):732–794, Firstquarter 2016. ISSN 1553-877X. doi: 10.1109/COMST.2015.2481183.

[44] Inès De Courchelle, Tom Guérout, Georges Da Costa, Thierry Monteil, and Yann Labit. Green energy efficient scheduling management. *Simulation Modelling Practice and Theory*, 93:208–232, May 2019. ISSN 1569-190X. doi: 10.1016/j.simpat.2018.09.011.

[45] Kalyanmoy Deb. Multi-objective optimization. In *Search Methodologies*, pages 403–449. Springer, 2014.

[46] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[47] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, pages 105–145. Springer, 2005.

[48] Apparao Dekka, Reza Ghaffari, Bala Venkatesh, and Bin Wu. A survey on energy storage technologies in power systems. In *Electrical Power and Energy Conference (EPEC), 2015 IEEE*, pages 105–111. IEEE, 2015.

[49] R. Deng, Z. Yang, M. Chow, and J. Chen. A Survey on Demand Response in Smart Grids: Mathematical Models and Approaches. *IEEE Transactions on Industrial Informatics*, 11(3):570–582, June 2015. ISSN 1551-3203. doi: 10.1109/TII.2015.2414719.

[50] W. Deng, F. Liu, H. Jin, and X. Liao. Online control of datacenter power supply under uncertain demand and renewable energy. In *2013 IEEE International Conference on Communications (ICC)*, pages 4228–4232, June 2013. doi: 10.1109/ICC.2013.6655227.

[51] X. Deng, D. Wu, J. Shen, and J. He. Eco-Aware Online Power Management and Load Scheduling for Green Cloud Datacenters. *IEEE Systems Journal*, 10(1):78–87, March 2016. ISSN 1932-8184. doi: 10.1109/JSYST.2014.2344028.

[52] S. Di, D. Kondo, and W. Cirne. Characterization and Comparison of Cloud versus Grid Workloads. In *2012 IEEE International Conference on Cluster Computing*, pages 230–238, September 2012. doi: 10.1109/CLUSTER.2012.35.

[53] S. Di, D. Kondo, and F. Cappello. Characterizing Cloud Applications on a Google Data Center. In *2013 42nd International Conference on Parallel Processing*, pages 468–473, October 2013. doi: 10.1109/ICPP.2013.56.

[54] Maimouna Diagne, Mathieu David, Philippe Lauret, John Boland, and Nicolas Schmutz. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, 27:65–76, November 2013. ISSN 1364-0321. doi: 10.1016/j.rser.2013.06.042.

[55] Z. Ding, L. Xie, Y. Lu, P. Wang, and S. Xia. Emission-Aware Stochastic Resource Planning Scheme for Data Center Microgrid Considering Batch Workload Scheduling and Risk Management. *IEEE Transactions on Industry Applications*, 54(6):5599–5608, November 2018. ISSN 0093-9994. doi: 10.1109/TIA.2018.2851516.

[56] D. Dirnberger. Photovoltaic module measurement and characterization in the laboratory. In Nicola Pearsall, editor, *The Performance of Photovoltaic (PV) Systems*, pages 23–70. Woodhead Publishing, January 2017. ISBN 978-1-78242-336-2. doi: 10.1016/B978-1-78242-336-2.00002-1.

[57] C. Draxl, B. M. Hodge, A. Clifton, and J. McCaa. Overview and Meteorological Validation of the Wind Integration National Dataset toolkit. Technical Report NREL/TP-5000-61740, National Renewable Energy Lab. (NREL), Golden, CO (United States), April 2015.

[58] S. A. Dudani. The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(4):325–327, April 1976. ISSN 0018-9472. doi: 10.1109/TSMC.1976.5408784.

[59] Bruce Dunn, Haresh Kamath, and Jean-Marie Tarascon. Electrical Energy Storage for the Grid: A Battery of Choices. *Science*, 334(6058):928–935, November 2011. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1212741.

[60] Matthias Ehrgott. *Multicriteria Optimization*. Springer Science & Business Media, May 2005. ISBN 978-3-540-21398-7.

[61] Linda Ager-Wick Ellingsen, Christine Roxanne Hung, and Anders Hammer Strømman. Identifying key assumptions and differences in life cycle assessment studies of lithium-ion traction batteries with focus on greenhouse gas emissions. *Transportation Research Part D: Transport and Environment*, 55:82–90, August 2017. ISSN 1361-9209. doi: 10.1016/j.trd.2017.06.028.

[62] Ben Elliston, Iain MacGill, and Mark Diesendorf. Least cost 100% renewable electricity scenarios in the Australian National Electricity Market. *Energy Policy*, 59: 270–282, August 2013. ISSN 0301-4215. doi: 10.1016/j.enpol.2013.03.038.

[63] N. A. Engerer and F. P. Mills. KPV: A clear-sky index for photovoltaics. *Solar Energy*, 105:679–693, July 2014. ISSN 0038-092X. doi: 10.1016/j.solener.2014.04.019.

[64] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. Power Provisioning for a Warehouse-sized Computer. In *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ISCA '07, pages 13–23, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-706-3. doi: 10.1145/1250662.1250665.

[65] Yanfeng Fan. Research on factors influencing an individual's behavior of energy management: A field study in China. *Journal of Management Analytics*, 4(3):203–239, July 2017. ISSN 2327-0012. doi: 10.1080/23270012.2017.1310000.

[66] Thomas F. Fuller, Marc Doyle, and John Newman. Simulation and optimization of the dual lithium ion insertion cell. *Journal of the Electrochemical Society*, 141(1): 1–10, 1994.

[67] Fréjus Gbaguidi, Selma Boumerdassi, Éric Renault, and Eugène Ezin. Characterizing servers workload in Cloud Datacenters. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference On*, pages 657–661. IEEE, 2015.

[68] M. Ghamkhari and H. Mohsenian-Rad. Energy and Performance Management of Green Data Centers: A Profit Maximization Approach. *IEEE Transactions on Smart Grid*, 4(2):1017–1025, June 2013. ISSN 1949-3053. doi: 10.1109/TSG.2013.2237929.

[69] Kenneth Gillingham, David Rapson, and Gernot Wagner. The Rebound Effect and Energy Efficiency Policy. *Review of Environmental Economics and Policy*, 10(1): 68–88, January 2016. ISSN 1750-6816. doi: 10.1093/reep/rev017.

[70] Íñigo Goiri, Kien Le, Md. E. Haque, Ryan Beauchea, Thu D. Nguyen, Jordi Guitart, Jordi Torres, and Ricardo Bianchini. GreenSlot : Scheduling Energy Consumption in Green Datacenters. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 20:1–20:11, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0771-0. doi: 10.1145/2063384.2063411.

[71] Íñigo Goiri, William Katsak, Kien Le, Thu D. Nguyen, and Ricardo Bianchini. Parasol and GreenSwitch: Managing Datacenters Powered by Renewable Energy. In *Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '13, pages 51–64,

New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1870-9. doi: 10.1145/2451116.2451123.

[72] B. Goss, I. R. Cole, E. Koubli, D. Palmer, T. R. Betts, and R. Gottschalg. Modelling and prediction of PV module energy yield. In Nicola Pearsall, editor, *The Performance of Photovoltaic (PV) Systems*, pages 103–132. Woodhead Publishing, January 2017. ISBN 978-1-78242-336-2. doi: 10.1016/B978-1-78242-336-2.00004-5.

[73] Léo Grange, Patricia Stolf, Georges Da Costa, and Amal Sayah. Heuristiques d'ordonnancement pour les centres de données alimentés par énergies renouvelables. In *Conférence d'informatique En Parallélisme, Architecture et Système*, Sophia Antipolis, France, June 2017.

[74] Léo Grange, Georges Da Costa, and Patricia Stolf. Green IT scheduling for data center powered with renewable energy. *Future Generation Computer Systems*, 86: 99–120, September 2018. ISSN 0167-739X.

[75] Léo Grange, Patricia Stolf, Georges Da Costa, and Paul Renaud-Goud. Cooperative and multi-objective power planning negotiation between datacenter and on-site renewable energy sources. Research Report IRIT/RR–2018–07–FR, IRIT, December 2018.

[76] Léo Grange, Patricia Stolf, Georges Da Costa, and Paul Renaud-Goud. Négociation multiobjectif de profils de puissance de centre de données alimenté par énergies renouvelables sur site. In *Conférence d'informatique En Parallélisme, Architecture et Système*, Toulouse, France, July 2018.

[77] Léo Grange, Patricia Stolf, Georges Da Costa, and Paul Renaud-Goud. Cooperative and Multi-Objective Power Planning Negotiation between Datacenter and on-Site Renewable Energy Sources. In *Submitted to IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, 2019.

[78] Chonglin Gu, Longxiang Fan, Wenbin Wu, Hejiao Huang, and Xiaohua Jia. Greening cloud data centers in an economical way by energy trading with power grid. *Future Generation Computer Systems*, 78(Part 1):89–101, January 2018. ISSN 0167-739X. doi: 10.1016/j.future.2016.12.029.

[79] Weijun Gu, Zechang Sun, Xuezhe Wei, and Haifeng Dai. A new method of accelerated life testing based on the Grey System Theory for a model-based lithium-ion battery life evaluation system. *Journal of Power Sources*, 267:366–379, December 2014. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2014.05.103.

[80] Tom Guérout, Samir Medjiah, Georges Da Costa, and Thierry Monteil. Quality of service modeling for green scheduling in Clouds. *Sustainable Computing: Informatics and Systems*, 4(4):225–240, December 2014. ISSN 2210-5379. doi: 10.1016/j.suscom.2014.08.006.

[81] C. W. Hansen and A. Pohl. Which models matter: Uncertainty and sensitivity analysis for photovoltaic power systems. In *2014 IEEE 40th Photovoltaic Specialist Conference (PVSC)*, pages 0175–0180, June 2014. doi: 10.1109/PVSC.2014.6925511.

[82] Han Hao, Zhexuan Mu, Shuhua Jiang, Zongwei Liu, and Fuquan Zhao. GHG Emissions from the Production of Lithium-Ion Batteries for Electric Vehicles in China. *Sustainability*, 9(4):504, April 2017. doi: 10.3390/su9040504.

[83] M. Hicks. Clank: Architectural support for intermittent computation. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 228–240, June 2017. doi: 10.1145/3079856.3080238.

[84] Karl Hinderer, Ulrich Rieder, and Michael Stieglitz. *Dynamic Optimization: Deterministic and Stochastic Models*. Universitext. Springer International Publishing, 2016. ISBN 978-3-319-48813-4.

[85] A. Hooshmand, M. H. Poursaeidi, J. Mohammadpour, H. A. Malki, and K. Grigoriads. Stochastic model predictive control method for microgrid management. In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–7, January 2012. doi: 10.1109/ISGT.2012.6175660.

[86] Y. Huang, S. Mao, and R. M. Nelms. Adaptive Electricity Scheduling in Microgrids. *IEEE Transactions on Smart Grid*, 5(1):270–281, January 2014. ISSN 1949-3053. doi: 10.1109/TSG.2013.2282823.

[87] Onyeka Illoh, Shaun Aghili, and Sergey Butakov. Using COBIT 5 for Risk to Develop Cloud Computing SLA Evaluation Templates. In Farouk Toumani, Barbara Pernici, Daniela Grigori, Djamal Benslimane, Jan Mendling, Nejib Ben Hadj-Alouane, Brian Blake, Olivier Perrin, Iman Saleh Moustafa, and Sami Bhiri, editors, *Service-Oriented Computing - ICSOC 2014 Workshops*, Lecture Notes in Computer Science, pages 236–249. Springer International Publishing, 2015. ISBN 978-3-319-22885-3.

[88] International Energy Agency. Key World Energy Statistics 2018. Technical report, 2018.

[89] Alexandru Iosup, Hui Li, Mathieu Jan, Shanny Anoep, Catalin Dumitrescu, Lex Wolters, and Dick HJ Epema. The grid workloads archive. *Future Generation Computer Systems*, 24(7):672–686, 2008.

[90] Ercan İzgi, Ahmet Öztopal, Bihter Yerli, Mustafa Kemal Kaymak, and Ahmet Duran Şahin. Short–mid-term solar power prediction by using artificial neural networks. *Solar Energy*, 86(2):725–733, February 2012. ISSN 0038-092X. doi: 10.1016/j.solener.2011.11.013.

[91] C. Jiang, Y. Wang, D. Ou, B. Luo, and W. Shi. Energy Proportional Servers: Where Are We in 2016? In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1649–1660, June 2017. doi: 10.1109/ICDCS.2017.285.

[92] Raúl Alberto Jiménez, Tomás Serebrisky, and Jorge Enrique Mercado Díaz. Power lost: Sizing electricity losses in transmission and distribution systems in Latin America and the Caribbean. 2014.

[93] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, January 2005. ISSN 1432-7643, 1433-7479. doi: 10.1007/s00500-003-0328-5.

[94] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python. http://www.scipy.org, 2001.

[95] M. R. Jongerden and B. R. Haverkort. Battery Modeling. Info:EuRepo/Semantics/Report TR-CTI, Centre for Telematics and Information Technology, University of Twente, Enschede, January 2008.

[96] Marijn R. Jongerden and Boudewijn R. Haverkort. Battery Aging, Battery Charging and the Kinetic Battery Model: A First Exploration. In Nathalie Bertrand and Luca Bortolussi, editors, *Quantitative Evaluation of Systems*, Lecture Notes in Computer Science, pages 88–103. Springer International Publishing, 2017. ISBN 978-3-319-66335-7.

[97] Azadeh Kamjoo, Alireza Maheri, Arash M. Dizqah, and Ghanim A. Putrus. Multi-objective design under uncertainties of hybrid renewable energy system using NSGA-II and chance constrained programming. *International Journal of Electrical Power & Energy Systems*, 74:187–194, January 2016. ISSN 0142-0615. doi: 10.1016/j.ijepes.2015.07.007.

[98] Marios K. Karakasis and Kyriakos C. Giannakoglou. On the use of metamodel-assisted, multi-objective evolutionary algorithms. *Engineering Optimization*, 38(8): 941–957, 2006.

[99] Deepak Paramashivan Kaundinya, P. Balachandra, and N. H. Ravindranath. Grid-connected versus stand-alone energy systems for decentralized power—A review of literature. *Renewable and Sustainable Energy Reviews*, 13(8):2041–2050, October 2009. ISSN 1364-0321. doi: 10.1016/j.rser.2009.02.002.

[100] S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan. An Analysis of Traces from a Production MapReduce Cluster. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, pages 94–103, May 2010. doi: 10.1109/CCGRID.2010.112.

[101] Atefeh Khosravi, Adel Nadjaran Toosi, and Rajkumar Buyya. Online virtual machine migration for renewable energy usage maximization in geographically distributed cloud data centers. *Concurrency and Computation: Practice and Experience*, page 0, May 2017. ISSN 1532-0634. doi: 10.1002/cpe.4125.

[102] Jungsoo Kim, Martino Ruggiero, David Atienza, and Marcel Lederberger. Correlation-aware Virtual Machine Allocation for Energy-efficient Datacenters. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, pages 1345–1350, San Jose, CA, USA, 2013. EDA Consortium. ISBN 978-1-4503-2153-2.

[103] Geoffrey T. Klise and Joshua S. Stein. Models used to assess the performance of photovoltaic systems. *Sandia National Laboratories*, 2009.

[104] Mykel J. Kochenderfer. *Decision Making under Uncertainty: Theory and Application*. Lincoln Laboratory Series. The MIT Press, Cambridge, Massachusetts, 2015. ISBN 978-0-262-02925-4.

[105] Fanxin Kong and Xue Liu. A Survey on Green-Energy-Aware Power Management for Datacenters. *ACM Computing Surveys*, 47(2):1–38, November 2014. ISSN 03600300. doi: 10.1145/2642708.

[106] V. Kostylev and A. Pavlovski. Solar power forecasting performance–towards industry standards. In *1st International Workshop on the Integration of Solar Power into Power Systems, Aarhus, Denmark*, 2011.

[107] Vincent Krakowski, Edi Assoumou, Vincent Mazauric, and Nadia Maïzi. Reprint of Feasible path toward 40–100% renewable energy shares for power supply in France by 2050: A prospective analysis. *Applied Energy*, 184:1529–1550, December 2016. ISSN 0306-2619. doi: 10.1016/j.apenergy.2016.11.003.

[108] Kyriakos Kritikos, Barbara Pernici, Pierluigi Plebani, Cinzia Cappiello, Marco Comuzzi, Salima Benrernou, Ivona Brandic, Attila Kertész, Michael Parkin, and Manuel Carro. A Survey on Service Quality Description. *ACM Comput. Surv.*, 46 (1):1:1–1:58, July 2013. ISSN 0360-0300. doi: 10.1145/2522968.2522969.

[109] K. Prakash Kumar and B. Saravanan. Recent techniques to model uncertainties in power generation from renewable energy sources and loads in microgrids – A review. *Renewable and Sustainable Energy Reviews*, 71:348–358, May 2017. ISSN 1364-0321. doi: 10.1016/j.rser.2016.12.063.

[110] N. Kumar, G. S. Aujla, S. Garg, K. Kaur, R. Ranjan, and S. K. Garg. Renewable Energy-Based Multi-Indexed Job Classification and Container Management Scheme for Sustainability of Cloud Data Centers. *IEEE Transactions on Industrial Informatics*, 15(5):2947–2957, May 2019. ISSN 1551-3203. doi: 10.1109/TII.2018.2800693.

[111] K. Kurowski, A. Oleksiak, W. Piątek, T. Piontek, A. Przybyszewski, and J. Węglarz. DCworms – A tool for simulation of energy efficiency in distributed computing infrastructures. *Simulation Modelling Practice and Theory*, 39:135–151, December 2013. ISSN 1569-190X. doi: 10.1016/j.simpat.2013.08.007.

[112] Jay H. Lee. Energy supply planning and supply chain optimization under uncertainty. *Journal of Process Control*, 24(2):323–331, February 2014. ISSN 0959-1524. doi: 10.1016/j.jprocont.2013.09.025.

[113] Hongtao Lei, Tao Zhang, Yajie Liu, Yabing Zha, and Xiaomin Zhu. SGEESS: Smart green energy-efficient scheduling strategy with dynamic electricity price for data center. *Journal of Systems and Software*, 108:23–38, October 2015. ISSN 0164-1212. doi: 10.1016/j.jss.2015.06.026.

[114] Hongtao Lei, Rui Wang, Tao Zhang, Yajie Liu, and Yabing Zha. A multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center. *Computers & Operations Research*, 75:103–117, November 2016. ISSN 0305-0548. doi: 10.1016/j.cor.2016.05.014.

[115] Chao Li, Rui Wang, Tao Li, Depei Qian, and Jingling Yuan. Managing Green Datacenters Powered by Hybrid Renewable Energy Systems. In *11th International Conference on Autonomic Computing ({ICAC} 14)*, pages 261–272, 2014.

[116] Shuhui Li, D. C. Wunsch, E. A. O'Hair, and M. G. Giesselmann. Using neural networks to estimate wind turbine power generation. *IEEE Transactions on Energy Conversion*, 16(3):276–282, September 2001. ISSN 0885-8969. doi: 10.1109/60.937208.

[117] Xiang Li, Xiaohong Jiang, Peter Garraghan, and Zhaohui Wu. Holistic energy and failure aware workload scheduling in Cloud datacenters. *Future Generation Computer Systems*, 78(Part 3):887–900, January 2018. ISSN 0167-739X. doi: 10.1016/j.future.2017.07.044.

[118] Y. Li, A. C. Orgerie, and J. M. Menaud. Opportunistic Scheduling in Clouds Partially Powered by Green Energy. In *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pages 448–455, December 2015. doi: 10.1109/DSDIS.2015.80.

[119] Y. Li, A. C. Orgerie, and J. M. Menaud. Balancing the Use of Batteries and Opportunistic Scheduling Policies for Maximizing Renewable Energy Consumption in a Cloud Data Center. In *2017 25th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 408–415, March 2017. doi: 10.1109/PDP.2017.24.

[120] Yuling Li, Xiaoying Wang, Peicong Luo, and Qingyi Pan. Thermal-Aware Hybrid Workload Management in a Green Datacenter towards Renewable Energy Utilization. *Energies*, 12(8):1494, January 2019. doi: 10.3390/en12081494.

[121] Zhongjin Li, Jidong Ge, Chuanyi Li, Hongji Yang, Haiyang Hu, Bin Luo, and Victor Chang. Energy cost minimization with job security guarantee in Internet data center. *Future Generation Computer Systems*, 73(Supplement C):63–78, August 2017. ISSN 0167-739X. doi: 10.1016/j.future.2016.12.017.

[122] Hao Liang and Weihua Zhuang. Stochastic Modeling and Optimization in a Microgrid: A Survey. *Energies*, 7(4):2027–2050, April 2014. doi: 10.3390/en7042027.

[123] Arnaud Liefooghe, Matthieu Basseur, Jérémie Humeau, Laetitia Jourdan, and El-Ghazali Talbi. On optimizing a bi-objective flowshop scheduling problem in an uncertain environment. *Computers & Mathematics with Applications*, 64(12):3747–3762, December 2012. ISSN 0898-1221. doi: 10.1016/j.camwa.2012.02.051.

[124] Bingwei Liu, Yinan Lin, and Yu Chen. Quantitative workload analysis and prediction using Google cluster traces. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference On*, pages 935–940. IEEE, 2016.

[125] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, August 1989. ISSN 1436-4646. doi: 10.1007/BF01589116.

[126] L. Liu, A. Chattopadhyay, and U. Mitra. Exploiting policy structure for solving MDPs with large state space. In *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2018. doi: 10.1109/CISS.2018.8362275.

[127] Xing Liu, Panwen Liu, Hongjing Li, Zheng Li, Chengming Zou, Haiying Zhou, Xin Yan, and Ruoshi Xia. Energy-aware Task Scheduling Strategies with QoS Constraint for Green Computing in Cloud Data Centers. In *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems*, RACS '18, pages 260–267, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5885-9. doi: 10.1145/3264746.3264792.

[128] Z. Liu and S. Cho. Characterizing Machines and Workloads on a Google Cluster. In *2012 41st International Conference on Parallel Processing Workshops*, pages 397–403, September 2012. doi: 10.1109/ICPPW.2012.57.

[129] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. Data center demand response: Avoiding the coincident peak via workload shifting and local generation. *Performance Evaluation*, 70(10):770–791, October 2013. ISSN 0166-5316. doi: 10.1016/j.peva.2013.08.014.

[130] Zhenhua Liu, Iris Liu, Steven Low, and Adam Wierman. Pricing Data Center Demand Response. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '14, pages 111–123, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2789-3. doi: 10.1145/2591971.2592004.

[131] T. Logenthiran, D. Srinivasan, A. M. Khambadkone, and H. N. Aung. Multi-Agent System (MAS) for short-term generation scheduling of a microgrid. In *2010 IEEE International Conference on Sustainable Energy Technologies (ICSET)*, pages 1–6, December 2010. doi: 10.1109/ICSET.2010.5684943.

[132] E. Lorenz, J. Hurka, D. Heinemann, and H. G. Beyer. Irradiance Forecasting for the Power Prediction of Grid-Connected Photovoltaic Systems. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2(1):2–10, March 2009. ISSN 1939-1404. doi: 10.1109/JSTARS.2009.2020300.

[133] P. Manganiello, M. Balato, and M. Vitelli. A Survey on Mismatching and Aging of PV Modules: The Closed Loop. *IEEE Transactions on Industrial Electronics*, 62 (11):7276–7286, November 2015. ISSN 0278-0046. doi: 10.1109/TIE.2015.2418731.

[134] James F. Manwell and Jon G. McGowan. Lead acid battery storage model for hybrid energy systems. *Solar Energy*, 50(5):399–405, May 1993. ISSN 0038-092X. doi: 10.1016/0038-092X(93)90060-2.

[135] Bill Marion. Comparison of predictive models for photovoltaic module performance. In *2008 33rd IEEE Photovoltaic Specialists Conference*, pages 1–6, May 2008. doi: 10.1109/PVSC.2008.4922586.

[136] R. Timothy Marler and Jasbir S. Arora. The weighted sum method for multi-objective optimization: New insights. *Structural and multidisciplinary optimization*, 41(6):853–862, 2010.

[137] R.T. Marler and J.S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, April 2004. ISSN 1615-1488. doi: 10.1007/s00158-003-0368-6.

[138] S. Z. Martínez and C. A. C. Coello. Combining surrogate models and local search for dealing with expensive multi-objective optimization problems. In *2013 IEEE Congress on Evolutionary Computation*, pages 2572–2579, June 2013. doi: 10.1109/CEC.2013.6557879.

[139] Christoph Marty and Rolf Philipona. The clear-sky index to separate clear-sky from cloudy-sky situations in climate research. *Geophysical Research Letters*, 27 (17):2649–2652, 2000.

[140] D.F. Menicucci. PVFORM - a new approach to photovoltaic system performance modeling. January 1985.

[141] John Miller, Lori Bird, Jenny Heeter, and Bethany Gorham. Renewable Electricity Use by the U.S. Information and Communication Technology (ICT) Industry. Technical Report NREL/TP-6A20-64011, National Renewable Energy Lab. (NREL), Golden, CO (United States), July 2015.

[142] I. S. Moreno, P. Garraghan, P. Townend, and J. Xu. Analysis, Modeling and Simulation of Workload Patterns in a Large-Scale Utility Cloud. *IEEE Transactions on Cloud Computing*, 2(2):208–221, April 2014. ISSN 2168-7161. doi: 10.1109/TCC.2014.2314661.

[143] Akinobu Murata, Hideaki Ohtake, and Takashi Oozeki. Modeling of uncertainty of solar irradiance forecasts on numerical weather predictions with the estimation of multiple confidence intervals. *Renewable Energy*, 117:193–201, March 2018. ISSN 0960-1481. doi: 10.1016/j.renene.2017.10.043.

[144] National Renewable Energy Laboratory. Solar Power Data for Integration Studies. https://www.nrel.gov/grid/solar-power-data.html, 2012.

[145] Nima Nikmehr, Sajad Najafi-Ravadanegh, and Amin Khodaei. Probabilistic optimal scheduling of networked microgrids considering time-based demand response programs under uncertainty. *Applied Energy*, 198:267–279, July 2017. ISSN 0306-2619. doi: 10.1016/j.apenergy.2017.04.071.

[146] T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multi-objective optimisation. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, volume 2, pages 878–885 Vol.2, December 2003. doi: 10.1109/CEC.2003.1299759.

[147] A. C. Orgerie, L. Lefèvre, and J. P. Gelas. Demystifying energy consumption in Grids and Clouds. In *Green Computing Conference, 2010 International*, pages 335–342, August 2010. doi: 10.1109/GREENCOMP.2010.5598295.

[148] Ali Pahlevan, Maurizio Rossi, Pablo G. Del Valle, Davide Brunelli, and David Atienza. Joint Computing and Electric Systems Optimization for Green Datacenters. In *Handbook of Hardware/Software Codesign*, pages 1163–1183. Springer, Dordrecht, 2017. ISBN 978-94-017-7266-2 978-94-017-7267-9. doi: 10.1007/978-94-017-7267-9_35.

[149] G. Parise and L. Parise. Electrical Distribution for a Reliable Data Center. *IEEE Transactions on Industry Applications*, 49(4):1697–1702, July 2013. ISSN 0093-9994. doi: 10.1109/TIA.2013.2256332.

[150] D. Paul, W. Zhong, and S. K. Bose. Demand Response in Data Centers Through Energy-Efficient Scheduling and Simple Incentivization. *IEEE Systems Journal*, 11 (2):613–624, June 2017. ISSN 1932-8184. doi: 10.1109/JSYST.2015.2476357.

[151] Jinqing Peng, Lin Lu, and Hongxing Yang. Review on life cycle assessment of energy payback and greenhouse gas emission of solar photovoltaic systems. *Renewable and Sustainable Energy Reviews*, 19:255–274, March 2013. ISSN 1364-0321. doi: 10.1016/j.rser.2012.11.035.

[152] Jean-Marc Pierson, Gwilherm Baudic, Stéphane Caux, Berk Celik, Georges Da Costa, Léo Grange, Marwa Haddad, Jérôme Lecuivre, Jean-Marc Nicod, Laurent Philippe, Veronika Rehn-Sonigo, Robin Roche, Gustavo Rostirolla, Amal Sayah,

Patricia Stolf, Minh-Thuyen Thi, and Christophe Varnier. DATAZERO: Datacenter With Zero Emission and Robust Management Using Renewable Energy. *IEEE Access*, 7:103209–103230, 2019. doi: 10.1109/ACCESS.2019.2930368.

[153] Andreas Poullikkas. A comparative overview of large-scale battery systems for electricity storage. *Renewable and Sustainable Energy Reviews*, 27:778–788, November 2013. ISSN 13640321. doi: 10.1016/j.rser.2013.07.017.

[154] Warren B Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, 2nd edition edition, 2011.

[155] Martin L. Puterman, editor. *Markov Decision Processes*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, USA, April 1994. ISBN 978-0-470-31688-7 978-0-471-61977-2. doi: 10.1002/9780470316887.

[156] Laura-Diana Radu. Determinants of Green ICT Adoption in Organizations: A Theoretical Perspective. *Sustainability*, 8(8):731, July 2016. doi: 10.3390/su8080731.

[157] Ahmad Rahmoun and Helmuth Biechl. Modelling of Li-ion batteries using equivalent circuit diagrams. *Electrical review, ISSN*, pages 0033–2097, 2012.

[158] Parthasarathy Ranganathan, Phil Leech, David Irwin, and Jeffrey Chase. Ensemble-level Power Management for Dense Blade Servers. In *Proceedings of the 33rd Annual International Symposium on Computer Architecture*, ISCA '06, pages 66–77, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 978-0-7695-2608-9. doi: 10.1109/ISCA.2006.20.

[159] Venkat Rao, Gaurav Singhal, Anshul Kumar, and Nicolas Navet. Battery model for embedded systems. In *18th International Conference on VLSI Design Held Jointly with 4th International Conference on Embedded Systems Design*, pages 105–110. IEEE, 2005.

[160] R. G. Regis. Evolutionary Programming for High-Dimensional Constrained Expensive Black-Box Optimization Using Radial Basis Functions. *IEEE Transactions on Evolutionary Computation*, 18(3):326–347, June 2014. ISSN 1089-778X. doi: 10.1109/TEVC.2013.2262111.

[161] Charles Reiss, John Wilkes, and Joseph L. Hellerstein. Google cluster-usage traces: Format+ schema. *Google Inc., White Paper*, pages 1–14, 2011.

[162] Charles Reiss, Alexey Tumanov, Gregory R Ganger, Randy H Katz, and Michael A Kozuch. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In *Proceedings of the Third ACM Symposium on Cloud Computing*, page 7. ACM, 2012.

[163] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Y. H. Katz, and Michael A. Kozuch. *Towards Understanding Heterogeneous Clouds at Scale: Google Trace Analysis*. 2012.

[164] Nery Riquelme, Christian Von Lücken, and Benjamin Baran. Performance metrics in multi-objective optimization. In *Computing Conference (CLEI), 2015 Latin American*, pages 1–11. IEEE, 2015.

[165] Gustavo Rostirolla. *Scheduling in Cloud Data Center Powered by Renewable Energy Only With Mixed Phases-Based Workload.* PhD Thesis, University of Toulouse, Toulouse, France, 2019.

[166] Gustavo Rostirolla, Léo Grange, Minh-Thuyen Thi, Patricia Stolf, Jean-Marc Pierson, Georges Da Costa, Gwilherm Baudic, Marwa Haddad, Ayham Kassab, Jean-Marc Nicod, Laurent Philippe, Veronika Rehn-Sonigo, Robin Roche, Berk Celik, Stéphane Caux, and Jérôme Lecuivre. Sizing and Management of Energy Sources for Green Datacenters with Renewable Energy. *submitted to Elsevier Renewable Energy*, 2019.

[167] Nikolaos V. Sahinidis. Optimization under uncertainty: State-of-the-art and opportunities. *Computers & Chemical Engineering*, 28(6):971–983, June 2004. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2003.09.017.

[168] Nicolas Schilling, Martin Wistuba, Lucas Drumond, and Lars Schmidt-Thieme. Hyperparameter Optimization with Factorized Multilayer Perceptrons. In Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, João Gama, Alípio Jorge, and Carlos Soares, editors, *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 87–103. Springer International Publishing, 2015. ISBN 978-3-319-23525-7.

[169] Manajit Sengupta, Yu Xie, Anthony Lopez, Aron Habte, Galen Maclaurin, and James Shelby. The National Solar Radiation Data Base (NSRDB). *Renewable and Sustainable Energy Reviews*, 89:51–60, June 2018. ISSN 1364-0321. doi: 10.1016/j.rser.2018.03.003.

[170] Songqing Shan and G. Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, March 2010. ISSN 1615-147X, 1615-1488. doi: 10.1007/s00158-009-0420-2.

[171] Navin Sharma, Sean Barker, David Irwin, and Prashant Shenoy. Blink: Managing Server Clusters on Intermittent Power. In *Proceedings of the Sixteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS XVI, pages 185–198, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0266-1. doi: 10.1145/1950365.1950389.

[172] L. Shi and K. Rasheed. A Survey of Fitness Approximation Methods Applied in Evolutionary Algorithms. In Yoel Tenne and Chi-Keong Goh, editors, *Computational Intelligence in Expensive Optimization Problems*, Adaptation Learning and Optimization, pages 3–28. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-10701-6. doi: 10.1007/978-3-642-10701-6_1.

[173] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5):637–646, October 2016. ISSN 2327-4662. doi: 10.1109/JIOT.2016.2579198.

[174] Junaid Shuja, Abdullah Gani, Shahaboddin Shamshirband, Raja Wasim Ahmad, and Kashif Bilal. Sustainable cloud data centers: A survey of enabling techniques and technologies. *Renewable and Sustainable Energy Reviews*, 62:195–214, 2016.

[175] Pierluigi Siano. Demand response and smart grids—A survey. *Renewable and Sustainable Energy Reviews*, 30:461–478, February 2014. ISSN 1364-0321. doi: 10.1016/j.rser.2013.10.022.

[176] G. Sideratos and N. D. Hatziargyriou. An Advanced Statistical Method for Wind Power Forecasting. *IEEE Transactions on Power Systems*, 22(1):258–265, February 2007. ISSN 0885-8950. doi: 10.1109/TPWRS.2006.889078.

[177] Radomir S. Stanković and Bogdan J. Falkowski. The Haar wavelet transform: Its status and achievements. *Computers & Electrical Engineering*, 29(1):25–44, January 2003. ISSN 0045-7906. doi: 10.1016/S0045-7906(01)00011-8.

[178] Bruce N. Stram. Key challenges to expanding renewable energy. *Energy Policy*, 96: 728–734, September 2016. ISSN 0301-4215. doi: 10.1016/j.enpol.2016.05.034.

[179] Florian Strunk. *An Analysis of Linux Boot Times*. PhD thesis, Chemnitz University of Technology, 2008.

[180] Hongyang Sun, Patricia Stolf, Jean-Marc Pierson, and Georges Da Costa. Energy-efficient and thermal-aware resource management for heterogeneous datacenters. *Sustainable Computing: Informatics and Systems*, 4(4):292–306, December 2014. ISSN 2210-5379. doi: 10.1016/j.suscom.2014.08.005.

[181] Hongyang Sun, Patricia Stolf, and Jean-Marc Pierson. Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing datacenters. *Future Generation Computer Systems*, 71(Supplement C):157–170, June 2017. ISSN 0167-739X. doi: 10.1016/j.future.2017.02.005.

[182] S. Talari, M. Yazdaninejad, and M. Haghifam. Stochastic-based scheduling of the microgrid operation including wind turbines, photovoltaic cells, energy storages and responsive loads. *Transmission Distribution IET Generation*, 9(12):1498–1509, 2015. ISSN 1751-8687. doi: 10.1049/iet-gtd.2014.0040.

[183] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[184] J. D. Ullman. NP-complete scheduling problems. *Journal of Computer and System Sciences*, 10(3):384–393, June 1975. ISSN 0022-0000. doi: 10.1016/S0022-0000(75)80008-0.

[185] Guido Urdaneta, Guillaume Pierre, and Maarten van Steen. Wikipedia workload analysis for decentralized hosting. *Computer Networks*, 53(11):1830–1845, July 2009. ISSN 1389-1286. doi: 10.1016/j.comnet.2009.02.019.

[186] Peter Van den Bossche, Frédéric Vergels, Joeri Van Mierlo, Julien Matheys, and Wout Van Autenboer. SUBAT: An assessment of sustainable battery technology. *Journal of Power Sources*, 162(2):913–919, November 2006. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2005.07.039.

[187] Ward Van Heddeghem, Sofie Lambert, Bart Lannoo, Didier Colle, Mario Pickavet, and Piet Demeester. Trends in worldwide ICT electricity consumption from 2007 to 2012. *Computer Communications*, 50:64–76, September 2014. ISSN 0140-3664. doi: 10.1016/j.comcom.2014.02.008.

[188] Joeri Van Mierlo, Peter Van den Bossche, and Gaston Maggetto. Models of energy sources for EV and HEV: Fuel cells, batteries, ultracapacitors, flywheels and engine-generators. *Journal of Power Sources*, 128(1):76–89, March 2004. ISSN 0378-7753. doi: 10.1016/j.jpowsour.2003.09.048.

[189] Kristof Van Moffaert, Madalina M. Drugan, and Ann Nowé. Hypervolume-Based Multi-Objective Reinforcement Learning. In Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 352–366. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-37140-0.

[190] David A. Van Veldhuizen and Gary B. Lamont. Evolutionary computation and convergence to a pareto front. In *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228, 1998.

[191] V. Villebonnet, G. D. Costa, L. Lefevre, J. M. Pierson, and P. Stolf. Dynamically Building Energy Proportional Data Centers with Heterogeneous Computing Resources. In *2016 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 217–220, September 2016. doi: 10.1109/CLUSTER.2016.34.

[192] V. Villebonnet, G. Da Costa, L. Lefevre, J. M. Pierson, and P. Stolf. Energy Aware Dynamic Provisioning for Heterogeneous Data Centers. In *2016 28th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 206–213, October 2016. doi: 10.1109/SBAC-PAD.2016.34.

[193] G. von Laszewski, L. Wang, A. J. Younge, and X. He. Power-aware scheduling of virtual machines in DVFS-enabled clusters. In *2009 IEEE International Conference on Cluster Computing and Workshops*, pages 1–10, August 2009. doi: 10.1109/CLUSTR.2009.5289182.

[194] Handing Wang, Markus Olhofer, and Yaochu Jin. A mini-review on preference modeling and articulation in multi-objective optimization: Current status and challenges. *Complex & Intelligent Systems*, 3(4):233–245, December 2017. ISSN 2198-6053. doi: 10.1007/s40747-017-0053-9.

[195] L. Wang, G. von Laszewski, J. Dayal, and F. Wang. Towards Energy Aware Scheduling for Precedence Constrained Parallel Tasks in a Cluster with DVFS. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, pages 368–377, May 2010. doi: 10.1109/CCGRID.2010.19.

[196] Weijia Wang and Michèle Sebag. Hypervolume indicator and dominance reward based multi-objective Monte-Carlo Tree Search. *Machine Learning*, 92(2):403–429, September 2013. ISSN 1573-0565. doi: 10.1007/s10994-013-5369-0.

[197] Martin Weiss, Juliane Haufe, Michael Carus, Miguel Brandão, Stefan Bringezu, Barbara Hermann, and Martin K. Patel. A Review of the Environmental Impacts of Biobased Materials. *Journal of Industrial Ecology*, 16(s1):S169–S181, 2012. ISSN 1530-9290. doi: 10.1111/j.1530-9290.2012.00468.x.

[198] X. Wu, X. Wang, and C. Qu. A Hierarchical Framework for Generation Scheduling of Microgrids. *IEEE Transactions on Power Delivery*, 29(6):2448–2457, December 2014. ISSN 0885-8977. doi: 10.1109/TPWRD.2014.2360064.

[199] A. A. Zainelabden, A. Ibrahim, D. Kliazovich, and P. Bouvry. On Service Level Agreement Assurance in Cloud Computing Data Centers. In *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*, pages 921–926, June 2016. doi: 10.1109/CLOUD.2016.0137.

[200] Hainan Zhang, Shuangquan Shao, Hongbo Xu, Huiming Zou, and Changqing Tian. Free cooling of data centers: A review. *Renewable and Sustainable Energy Reviews*, 35:171–182, July 2014. ISSN 1364-0321. doi: 10.1016/j.rser.2014.04.017.

[201] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, March 2011. ISSN 22106502. doi: 10.1016/j.swevo.2011.03.001.

[202] Wei Zhou, Hongxing Yang, and Zhaohong Fang. A novel model for photovoltaic array performance prediction. *Applied Energy*, 84(12):1187–1198, December 2007. ISSN 03062619. doi: 10.1016/j.apenergy.2007.04.006.

[203] Zhou Zhou, Jemal Abawajy, Morshed Chowdhury, Zhigang Hu, Keqin Li, Hongbing Cheng, Abdulhameed A. Alelaiwi, and Fangmin Li. Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms. *Future Generation Computer Systems*, August 2017. ISSN 0167-739X. doi: 10.1016/j.future.2017.07.048.

[204] Ziming Zhu, Jie Tang, S. Lambotharan, Woon Hau Chin, and Zhong Fan. An integer linear programming based optimization for home demand-side management in smart grid. In *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–5, January 2012. doi: 10.1109/ISGT.2012.6175785.

[205] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999. ISSN 1089-778X. doi: 10.1109/4235.797969.

[206] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

[207] Eckart Zitzler, Marco Laumanns, Lothar Thiele, et al. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. TIK-report, 2001.